File No. 7090-36
Form C28-6248-7

IBM Systems Reference Library

IBM 7090/7094 IBSYS Operating System

Version 13

System Monitor (IBSYS)

This publication describes the use and general function of the System Monitor of the IBM 7090/7094 IBSYS Operating System, version 13 (7090-PR-130). The IBSYS Operating System consists of a number of commercial and scientific programming aids operating under overall control and direction of the System Monitor (7090-SV-918). It is designed to process sequentially a variety of unrelated jobs with little or no operator intervention.

The System Monitor includes the System Supervisor, the System Nucleus, the Input/Output Executor, the System Core-Storage Dump Program, and the System Editor. In general, the "Introduction" of this publication and the sections describing the System Supervisor and the System Core-Storage Dump Program are directed primarily to the applications programmer. The remaining sections are directed to the systems programmer.

Preface

This publication describes the use and general function of the System Monitor of the IBM 7090/7094 IBSYS Operating System (Version 13) and provides information for maintaining the system. The IBSYS Operating System may be considered an integral part of the IBM 7090/7094 Data Processing System. It consists of a comprehensive set of programming aids operating as subsystems under a master System Monitor.

The System Monitor described in this publication encompasses the System Supervisor, the System Nucleus, the Input/Output Executor, the System Core-Storage Dump Program, and the System Editor. The subsystems operating under the System Monitor are described in separate publications. These publications are referred to in the "Introduction" section of this manual and are listed in Figure 2 in the "System Supervisor" section. Instructions for the operator of the system are provided in a separate publication entitled *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

This publication on the System Monitor is divided into seven major sections. The first three sections, "Introduction," "System Supervisor," and "System Core-Storage Dump Program," are directed primarily to the applications programmer. The remaining sections are directed primarily to the systems programmer. The systems programmer is an experienced programmer who is assigned to place the IBSYS Operating System into operation, modify it according to the special requirements of his installation, maintain it, and ensure adequate control over its content and use.

All of the System Monitor control cards that might be used by the applications programmer in programming a job are described in the section on the System Supervisor. However, very few of the cards are actually required for most jobs. Therefore, these cards are described first, for the benefit of the reader who is only interested in the control cards required to run an average job such as a FORTRAN compilation and execution. The control cards described in the remainder of the section are primarily of interest to the more experienced programmer, the systems programmer, and the operator.

The reader of this manual is assumed to be familiar with the contents of the IBM reference manual *IBM 7094 Data Processing System*, Form A22-6703.

EIGHTH EDITION

This is a reprint of C28-6248-6 incorporating changes released in the following Technical Newsletter:

FORM NUMBER	PAGES AFFECTED	DATE
N28-0207	29 and 30	December 30, 1966

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

Address comments concerning the contents of the publication to IBM Corporation, Programming Publications, 1271 Avenue of the Americas, New York, N.Y. 10020.

Contents

Introduction

Function

The 7090/7094 IBSYS Operating System consists of a comprehensive set of commercial and scientific programming aids operating as subsystems, under executive control and coordination of a System Monitor. The Monitor, by coordinating the operation of the subsystems, enables a series of unrelated jobs to be processed with little or no operator intervention. By reducing the degree of human participation in the mechanics of data processing, the Operating System ensures that jobs are processed faster, more efficiently, and are less subject to human error. As a result, turnaround time, the interval between the time a programmer submits a job for processing and the time he receives results, is significantly reduced.

Organization

System Monitor

The general organization of the System Monitor, as well as its general relation to the subsystems operating under it, is illustrated in Figure 1. The System Monitor consists of:

1. The System Supervisor, whose primary function is to control and coordinate the processing of jobs by passing control from one subsystem to another.
2. The System Nucleus, which remains in core storage at all times and provides common facilities for intercommunication and control among the subsystems and between the System Monitor and the subsystems.
3. The Input/Output Executor, which normally remains in core storage to coordinate and control input/output and other trapping operations.
4. The System Core-Storage Dump Program, which may be used to facilitate the testing and analysis of any program executed under control of the Operating System.
5. The System Editor, which provides the systems programmer with facilities for modifying and maintaining the System Monitor and the subsystems operating under it.

The System Monitor may also contain an installation accounting routine tailored to the specific requirements of the installation.

The subsystems operating under the System Monitor (Figure 1) provide the programmer with a variety of programming facilities which he may use singly or in combination to process a particular job. Each of these subsystems is described briefly below.

IBJOB Processor

The IBJOB Processor is the major subsystem of the IBSYS Operating System. It is a highly integrated processor which consists of the following:

The Processor Monitor (IBJOB)
The FORTRAN IV Compiler (IBFTC)
The COBOL Compiler (IBCBC)
The Assembler—The Macro Assembly Program (IBMAP)
A relocatable loader—The Loader (IBLDR)
Preassembled subroutines to be used, if required by the object program—The Subroutine Library (IBLIB)
The Debugging Package (IBDBL and IBDBC)

Just as the System Monitor (IBSYS) provides over-all control for the subsystems under it, the IBJOB Processor Monitor provides control over the IBJOB components. The compilers and the assembler under the control of the Processor Monitor are used to translate source programs, written in higher-level programming languages, into machine language object programs. The Debugging Package provides the programmer with a means of obtaining highly selective dynamic dumps of any core storage locations and any registers at selected points in his program. The Subroutine Library includes a complete Input/Output Control System. The Loader loads object programs in preparation for execution, and also loads the portions of IOCS that are required by that object program.

The IBJOB Processor and its use are described in detail in the publication *IBM 7090/7094 IBSYS Operating System: IBJOB Processor*, Form C28-6389.

FORTRAN II Processor

The FORTRAN II Processor can be used in either of two modes, the FORTRAN mode or the IBSFAP mode.

In the FORTRAN mode, the FORTRAN II Processor can be used to compile, assemble, load, and execute source programs written in FORTRAN II language. It can also assemble, load, and execute programs written in FORTRAN II Assembly Program (FAP) language and load and execute previously assembled object programs. Facilities are provided for combining program segments written in FORTRAN II and FAP languages with previously assembled segments to form a single executable object program. Facilities are also provided for chaining core storage loads so that executed portions of a program may be overlaid with portions yet to be executed.

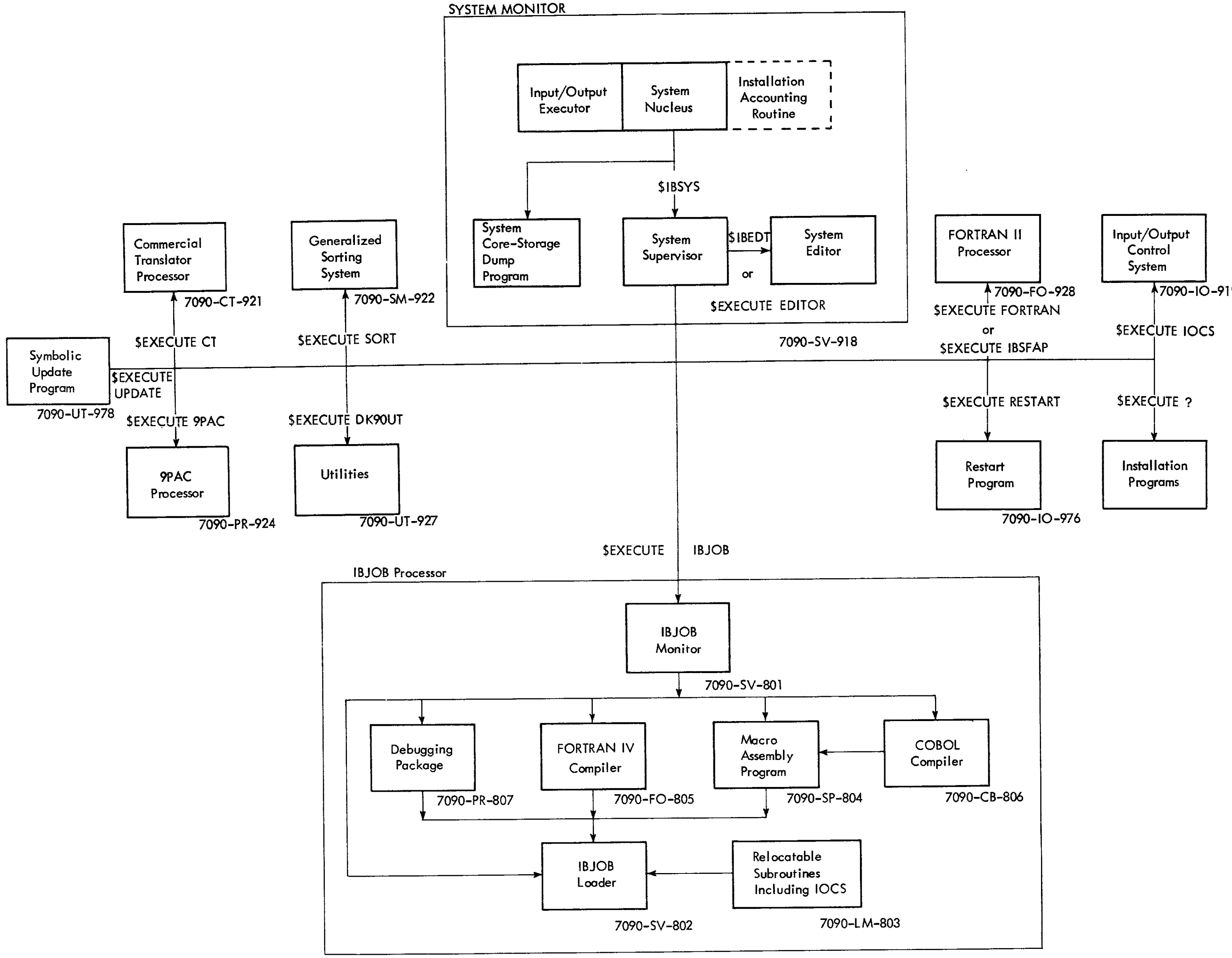


Figure 1. IBM 7090/7094 IBSYS Operating System

In the IBSFAP mode, the FORTRAN II Processor can be used to assemble, but not load and execute, programs written in FAP language. An assembled object program can be loaded and executed under control of the Input/Output Control System or the FORTRAN mode of the FORTRAN II Processor. The IBSFAP mode of the FORTRAN II Processor can also be used to update symbolic tapes by changing, deleting, or adding instructions.

The FORTRAN II Processor and its use are described in detail in the publication *IBM 7090/7094 Programming Systems: FORTRAN II Operations,* Form C28-6066.

The FORTRAN II Assembly Program and its use are described in the publication *IBM 7090/7094 Programming Systems: FORTRAN II Assembly Program (FAP),* Form C28-6235.

Input/Output Control System

The Input/Output Control System (Full IOCS) provides input/output control for programs assembled by the FORTRAN II Processor. It relieves the programmer of the task of writing complex input/output routines by automatically controlling the blocking and unblocking of data records, the overlapping of processing with input and output, and the preparation and checking of labels. Only those portions of IOCS actually required are loaded with the assembled object program.

IOCS and its use are described in detail in the publication *IBM 7090/7094 IBSYS Operating System: Input/Output Control System,* Form C28-6345.

Symbolic Update Program

The Symbolic Update Program can be used to modify symbolic tapes by changing, deleting, or adding symbolic card images and producing new symbolic tapes. This subsystem enables programmers to update programs already on tape in the MAP, FORTRAN IV, or COBOL languages. It can also be used to maintain multireel input and output tapes, extract card images from an input tape, space tape, and check the sequence of serialization of card images on tape.

The Symbolic Update Program and its use are described in the publication *IBM 7090/7094 IBSYS Operating System: Symbolic Update Program,* Form C28-6386.

Restart Program

Unlike the other subsystems operating under the System Monitor, the Restart Program is used exclusively by the operator of the system. It is designed to enable the operator to restart an interrupted program using a checkpoint record recorded by IOCS before the interruption occurred.

The use of the Restart Program is described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide,* Form C28-6355.

Commercial Translator Processor

The Commercial Translator Processor may be used to compile, assemble, load, and execute programs written in the IBM Commercial Translator language.

The Commercial Translator Processor and its use are described in the publication *IBM 709/7090 Commercial Translator Processor,* Form J28-6169.

Generalized Sorting System

The Generalized Sorting System can be used to sort fixed-length or variable-length records, written in either binary or decimal mode. The control fields of the records may be signed or unsigned. The records can be sorted in ascending or descending order, using either the commercial or scientific collating sequence.

The Generalized Sorting System and its use are described in the publication *IBM 7090/7094 Generalized Sorting System: 7090/7094 Sort,* Form C28-6365.

9PAC Processor

The 9PAC Processor can be used to establish and maintain data files and to generate reports on the data in the files.

The 9PAC Processor and its use are described in the following publications:

IBM 7090 Programming Systems, Share 7090 9PAC
- *Part 1: Introduction and General Principles,* Form J28-6166
- *Part 2: The File Processor,* Form J28-6167
- *Part 3: The Reports Generator,* Form J28-6168

Utilities

The Utilities consist of a tape dump routine for 729 Magnetic Tape Units and 7340 Hypertape Drives and of the following for 1301/2302 Disk Storage and 7320 Drum Storage: format track generation, home address and record address identification, load disk/drum, dump disk/drum, dump 2302 disk platter, restore disk/drum, and clear disk/drum.

The utilities and their use are described in the following publication:

IBM 7090/7094 IBSYS Operating System: Utilities, Form C28-6364

User Programs

In addition to the subsystems described above, the user of the IBSYS Operating System may design programs and incorporate them as subsystems operating under the System Monitor. Conversely, the user may remove subsystems or portions of subsystems that are not required at his installation.

Application

In programming a job, the programmer may employ any logical combination of the subsystems operating under the System Monitor. The programmer, in effect, controls and directs the processing of his job from his desk by inserting the proper control cards in his job deck. Before a particular job is processed, it may be combined with other jobs to form a single input file of unrelated jobs. The input file of jobs can then be processed by the Operating System without costly setup delays between jobs or job segments while the data processing system lies idle.

The operator of the system, for the most part, performs relatively routine functions, such as loading or unloading tape reels. Usually he is told what to do and when by means of an on-line printout from the Operating System. If the Operating System, owing to a programmer error, cannot complete a job or job segment, it automatically skips to the next job or job segment without intervention by the operator. However, the operator can, if he chooses, interrupt the Operating System at the end of any job. By means of control cards, he can then direct the Operating System to perform any one of several operations, for example, to restart at the beginning of another job on the input file.

If an error occurs during the execution of an object program, the operator, the Operating System, or the object program itself can call for a post-mortem dump of any part or all of the core storage (to facilitate analysis of the error) followed by an automatic skip to the next job segment. An object program can also, at any point in the program, call for a snap dump of any part or all of core storage. At the completion of the snap dump, the contents of core storage are restored and the execution of the object program is resumed. Any one of six formats can be specified for a core storage dump. (See Figure 6 in the section "System Core-Storage Dump Program.")

When an input file of jobs is completed, the Operating System stops after providing the operator with information on the status of the system input and output files. Then, by the use of control cards, the operator can direct the Operating System to perform any one of a number of operations. For example, he may direct it to restart at the beginning of a new input file or to rewind and unload the input and output files.

System Unit Functions

To ensure continuous job processing and proper coordination between subsystems, the System Monitor provides a logical framework for assigning input/output units to specific functions and for keeping track of the exact status of all units at all times. Some units are assigned system unit functions, that is, they are assigned specific functions required by the nature of the Operating System and may be used in that capacity by the System Monitor and each of the subsystems operating under it. For example, at least one unit is used as a system input file on which jobs are stacked so that they can be processed continuously by the System Monitor and the various subsystems. Any units not assigned to system unit functions are available for use by the programmer, provided they are not logically detached from the Operating System. Appendix F contains charts showing which of the system unit functions are required by the Operating System and by each of its subsystems, as well as the types of devices that may be assigned to these functions.

The following are the system unit functions to which units may be assigned. Some functions may not have units assigned to them, depending on the requirements of a particular installation.

Information as to which types of input/output devices can be assigned to specific system functions for a particular subsystem may be found in the *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355, under "Use of Input/Output Units."

System Input

A System Input Unit is required by every installation for use as a common job input file. Normally, the input file tape is prepared off-line on an auxiliary 1401 Data Processing System. The input file may contain System Monitor and subsystem control cards, symbolic source programs, binary object programs, and data.

System Output

A System Output Unit is required by every installation for use as a common output print file. The output file will contain messages from the Operating System and may contain source program listings, assembly listings, core storage dump listings, and output data.

System Peripheral Punch

A unit must be assigned as a System Peripheral Punch Unit. The peripheral punch file may be processed off-line to produce binary object program card decks.

The same unit can be assigned as both the System Output Unit and the System Peripheral Punch Unit. However, when this dual assignment is used, FORTRAN II programs cannot be executed. In addition, some output may be lost when the combined unit is backspaced to suppress punched output due to errors in relocatable FAP or IBSFAP assemblies.

System Library

At least one unit must be assigned as a System Library Unit on which the IBSYS Operating System itself is recorded. Up to four units may be assigned as library units. When magnetic tape units are used, the Operating System may be duplicated on two System Library Tapes which can be referred to alternately in order to reduce delays in processing while the library tape is being rewound. The Operating System can also be split between two or more library units in order to reduce the time required to gain access to particular parts of the Operating System.

System Utility

Four System Utility Units are required by every installation for general use by the System Monitor and the subsystems operating under it. Up to five additional units may be assigned to System Utility functions for object program use. The required System Utility Units may also be used by object programs. However, the programmer should avoid assigning any files that are to be retained, to System Utility Units that will be used by the Operating System during the job.

A unit assigned as a System Utility Unit may be simultaneously assigned as a System Checkpoint Unit.

System Checkpoint

A unit may be assigned as a System Checkpoint Unit on which checkpoint records are recorded under IOCS control. The same unit may also be assigned as a System Utility Unit.

System Printer

A 716 Printer is assigned as the *System Printer*. It is used by the Operating System to record messages to the operator.

System Card Reader

A 711 Card Reader is assigned as the System Card Reader. Normally, the card reader is used by the operator for inserting control cards that direct the Operating System. The card reader may also be used as a substitute input unit for processing small job files.

System Card Punch

A 721 Card Punch may be assigned as the System Card Punch. The punch may be used to punch control cards for use by the operator, or it may be used as a substitute output unit.

Alternate Units

A second unit may be assigned to the system input, output, peripheral punch, or checkpoint functions. If a second unit is assigned, it serves as an alternate unit to eliminate delays due to reel switching. When an end of tape is reached on one unit an automatic switch can be made to the alternate unit. In the distributed version of the Operating System, the same 729 Magnetic Tape Unit is assigned to a system utility function (SYSUT4) and to the alternate peripheral punch function (SYSPP2). Therefore, if automatic reel switching is expected for the peripheral punch, different units should be assigned to these functions.

Machine Requirements

The following minimum machine configuration is required for use of the 7090/7094 IBSYS Operating System:

1. An IBM 7090/7094 Data Processing System.
2. Three IBM 729 Magnetic Tape Units or IBM 7340 Hypertape Drives for assignment as System Input, Output, and Peripheral Punch Units. A single unit may be assigned as a combined System Output and Peripheral Punch Unit (but note restrictions stated under "System Peripheral Punch"). If this is done, the extra unit may be assigned as a second System Library Unit.
3. Five units; one for assignment as a System Library Unit and four for assignment as System Utility Units. These units may be any combination of IBM 729 Magnetic Tape Units, 7340 Hypertape Drives, or selected cylinders of direct access storage units. (In the remainder of this publication, direct access storage, when used, will refer to both IBM 1301/2302 Disk Storage and IBM 7320 Drum Storage.) If more than one type of direct access storage unit is attached to the same 7909 data channel switch (interface) setting within a channel, the module numbers must be assigned in the following order:

   1301 modules <2302 modules <7320 modules
4. One IBM 716 Printer for assignment as the System Printer.
5. One IBM 711 Card Reader for assignment as the System Card Reader.

The 1401/1460 peripheral input/output programs for the IBM 7090/7094 IBSYS Operating System require an IBM 1401 Data Processing System with the following features:

1. 4000 locations of core storage
2. Advanced Programming feature
3. High-Low-Equal Compare feature
4. Sense Switch feature
5. One IBM 729 or 7330 Magnetic Tape Unit
6. One IBM 1402 Card Read Punch with the Read-Punch Release Feature and the Column Binary Feature (on 1401) or with the Binary Transfer and Bit Test Features (on 1460)
7. One 1403 Printer with 132 print positions and the Print Control feature

Additional information on the 1401/1460 peripheral input/output programs can be found in the publication IBM *7090/7094 IBSYS Operating System, Version 13: Operator's Guide*, Form C28-6355.

System Supervisor

Function

The primary function of the System Supervisor is to coordinate and supervise the processing of jobs by:

1. Passing control from one subsystem to another.
2. Restoring unit assignments between jobs.
3. Controlling interruption by the operator.
4. Skipping jobs or job segments when directed by a subsystem.

In addition, the System Supervisor can be directed by control cards to perform a variety of other functions, such as changing unit assignments, manipulating tape units, and passing control to the System Editor. The System Supervisor is directed to perform its functions mainly by means of control cards which it reads from a system input file, interprets, and acts upon.

Definition of Job and Job Segment

Of the many control cards that are recognized by the System Supervisor, the key cards in controlling the continuous processing of jobs are the $JOB card, $EXECUTE card, $IBSYS card, and $STOP card. Each of these cards is recognized and acted upon by each of the subsystems operating under control of the System Supervisor, as well as by the System Supervisor.

A job consists of all of the cards beginning with a $JOB card and ending with, but not including, the next $JOB card. Each job in a stack of jobs on an input file is considered to be entirely independent of any other job.

A $JOB card may be followed by a $EXECUTE card, $IBSYS card, or $ID card. If a $ID card, which is an optional card used for intrajob accounting purposes, follows the $JOB card, it must itself be followed by either a $EXECUTE or $IBSYS card.

A job segment to be performed by the System Monitor consists of all cards beginning with a $IBSYS card and ending with, but not including, the next $EXECUTE card, $JOB card, or $STOP card.

A job segment to be performed by a subsystem consists of all cards beginning with a $EXECUTE card and ending with, but not including, the next $EXECUTE card, $IBSYS card, $JOB card, or $STOP card. The job segment may consist of one or more applications of the particular subsystem specified on the $EXECUTE card. Any cards in the job segment that follow the $EXECUTE card are read and interpreted by the subsystem specified on the $EXECUTE card. They, therefore, must conform to the requirements of the specific subsystem. The publications describing the requirements for each subsystem are listed in Figure 2.

Control Card Format

The general format of the System Supervisor control cards follows:

COLUMNS	CONTENTS
1	$
2-8	Control card name left-justified
16-72	Variable field information (argument 1, argument 2,... ,argument n)

Columns 7 and 8 are not examined by the System Supervisor.

Embedded blanks are not allowed within arguments, except for the $DATE card. A comma separates arguments and a blank separates the last argument from comments.

In this publication, the following conventions are used for variable field information:

1. Lower-case letters indicate that a substitution must be made.
2. Upper-case letters must be punched exactly as shown.
3. Brackets [ ] contain an option that may be omitted or included by the user.
4. Braces { } indicate that a choice of the contents is mandatory.
5. A number over the first character of a field indicates the first card column of the field.

Basic Control Cards

In a typical installation, only the following System Supervisor control cards are normally required:

$JOB
$IBSYS
$EXECUTE
$STOP

These cards are recognized by each of the subsystems and the System Editor as well as by the System Supervisor. Detailed descriptions of the cards follow.

$JOB Card

Format:

```
1           16
$JOB        any text
```

This card defines the beginning of a job. It causes a transfer to the installation accounting routine (if one

Subsystem Name Specified on $EXECUTE Card	Full Name of Subsystem	Publications Describing Subsystems
IBJOB	IBJOB Processor	IBM 7090/7094 IBSYS Operating System: IBJOB Processor, Form C28-6389
FORTRAN	FORTRAN II Processor (FORTRAN Mode)	IBM 7090/7094 Programming Systems: FORTRAN II Operations, Form C28-6066
IBSFAP	FORTRAN II Processor (IBSFAP Mode)	IBM 7090/7094 Programming Systems: FORTRAN Assembly Program (FAP), Form C28-6235
IOCS	Input/Output Control System	IBM 7090/7094 IBSYS Operating System: Input/Output Control System, Form C28-6345
RESTART	Restart Program	IBM 7090/7094 IBSYS Operating System: Operator's Guide, Form C28-6355
CT	Commercial Translator Processor	IBM 7090/7094 Commercial Translator Processor, Form J28-6169
SORT	Generalized Sorting System	IBM 7090/7094 Generalized Sorting System: 7090/7094 Sort, Form C28-6365
9PAC	9PAC	IBM 7090 Programming Systems, Share 7090 9PAC Part 1: Introduction and General Principals, Form J28-6166 Part 2: The File Processor, Form J28-6167 Part 3: The Reports Generator, Form J28-6168
DK90UT	Utilities	IBM 7090/7094 IBSYS Operating System: Utilities, Form C28-6364
UPDATE	Symbolic Update Program	IBM 7090/7094 IBSYS Operating System: Symbolic Update Program, Form C28-6386
EDITOR	*System Editor	Described in this publication

*Although the System Editor is a part of the System Monitor itself, it acts like a subsystem in that it can be called by means of a $EXECUTE card.

Figure 2. Subsystems Operating Under Control of the System Monitor

exists at the installation) and the restoration of any units that were reassigned or made unavailable during a previous job, with the exception of the following:

1. Any unit that was logically detached from control by the Operating System.

2. Any unit that was assigned to a system unit function in place of a detached unit.

3. Any unit that was assigned to a system input, system output, or system peripheral punch function.

A unit originally assigned to a system input, system output, or system peripheral punch function is not restored because it may have been validly replaced by its alternate unit when an end of tape was encountered.

When a $JOB card is read by a subsystem, the System Supervisor is called into core storage only if it is required either to restore the status of a unit or to control a manually initiated between-jobs interrupt condition.

The $JOB card is listed on both the System Printer and the System Output Unit. Columns 16 through 72 are normally used to identify a job and may contain any combination of alphameric characters and blanks.

$IBSYS Card

Format:

```
1              16
$IBSYS
```

When this card is read by a subsystem or by the System Editor, the System Supervisor is called into core storage and control is relinquished to it. The System Supervisor then reads and processes succeeding control cards until control is relinquished to a subsystem by means of a $EXECUTE card or to the System Editor by means of a $EXECUTE EDITOR or $IBEDT card.

$EXECUTE Card

Format:

```
1              16
$EXECUTE       subsystem name
```

This card defines the beginning of a job segment that is to be processed by the specified subsystem. If the $EXECUTE card is read by the System Supervisor, the System Supervisor positions the proper System Library Unit to the specified subsystem, reads in the first record of the subsystem, and relinquishes control to it. If the card is read by a subsystem other than the one specified, control, as well as the subsystem name, is passed to the System Supervisor, which in turn, reads in the first record of the specified subsystem and relinquishes control to it. If the card is read by the specified subsystem, the subsystem retains control and proceeds to process the job segment.

The subsystem name consists of six or fewer BCD characters corresponding to a name in the System Name Table of the System Supervisor. The System Name Table indicates the arrangement of the participating subsystems on the System Library Units and is used by the System Supervisor to locate a subsystem specified by a $EXECUTE card.

Although the System Editor is an integral part of the System Monitor, it is recognized as a subsystem by the System Supervisor and can be called by a $EXECUTE EDITOR card as well as by the $IBEDT card. The name EDITOR is placed in the IBSYS System Name Table (see "$EXECUTE Card"), and the Editor respects the rules governing job-control communication.

The names for the subsystems provided with the distributed versions of the IBSYS Operating System are listed in Figure 2, together with the full name of each subsystem and the publication or publications describing it.

$STOP Card

Format:

```
1              16
$STOP
```

This card is used to define the end of a stack of jobs. It is usually placed, by the operator or a job setup man, at the end of a stack of jobs on the system input file. When the card is read by a subsystem, the System Supervisor is called into core storage and alerted to the fact that the card was read. Upon recognition of the $STOP card, the System Supervisor performs the following actions:

1. Prints, on the System Printer, the physical unit assignment and tape position (record and file count) of the System Output Unit, the System Peripheral Punch Unit, and the System Input Unit followed by the message:

   END OF JOBS

2. Writes a trailer label on the System Peripheral Punch Unit, if it is not at load point, and backspaces over the trailer label.

3. If a $REWIND SYSOU1, $REMOVE SYSOU1, or another $STOP card was not read previously*, writes a trailer label on the System Output Unit and backspaces over the trailer label.

4. Stops the machine with the System Card Reader temporarily assigned to the system input function.

When START is pressed, the control cards, if any, in the card reader are read and processed by the System Supervisor until they are depleted. When the cards in the card reader are depleted, the System Supervisor proceeds to read control cards from the unit assigned as the System Input Unit. Therefore, when the machine stops at the end of a stack of jobs, the operator may terminate job processing by using any of the System Supervisor control cards, such as the $ENDFILE and $REMOVE cards; or he may continue processing a new stack of jobs, either on the card reader or on the tape unit assigned as the System Input Unit.

Typical Input Deck

Figure 3 shows a composite system input deck containing several jobs.

Control Cards Used Primarily by the Experienced Programmer and the Operator

The control cards described in this section are mainly of interest to the experienced programmer and the operator. These control cards are recognized by the System Supervisor only and not by a subsystem. Therefore, they should be used only under the following conditions:

1. At the beginning of an input file following an initial start.

2. Following a $IBSYS card or other System Supervisor control cards, other than a $IBEDT card, that follow an $IBSYS card.

3. In the System Card Reader during a between-jobs or end-of-jobs interrupt.

Initialization Control Cards

$DATE Card

Format:

```
1              16
$DATE          mmddyy
```

This card is normally used by the operator at the beginning of each day. The card causes the six characters in columns 16 through 21 of the card to be stored in the SYSDAT word of the Communication Region of the System Nucleus. Although a subsystem may dis-

*After a $REWIND SYSOU1, $REMOVE SYSOU1, or $STOP card is read and processed, further use of the System Output Unit by the System Supervisor is suspended until a $JOB card is read or the System Supervisor is called into core storage again by a $IBSYS card or by a subsystem.

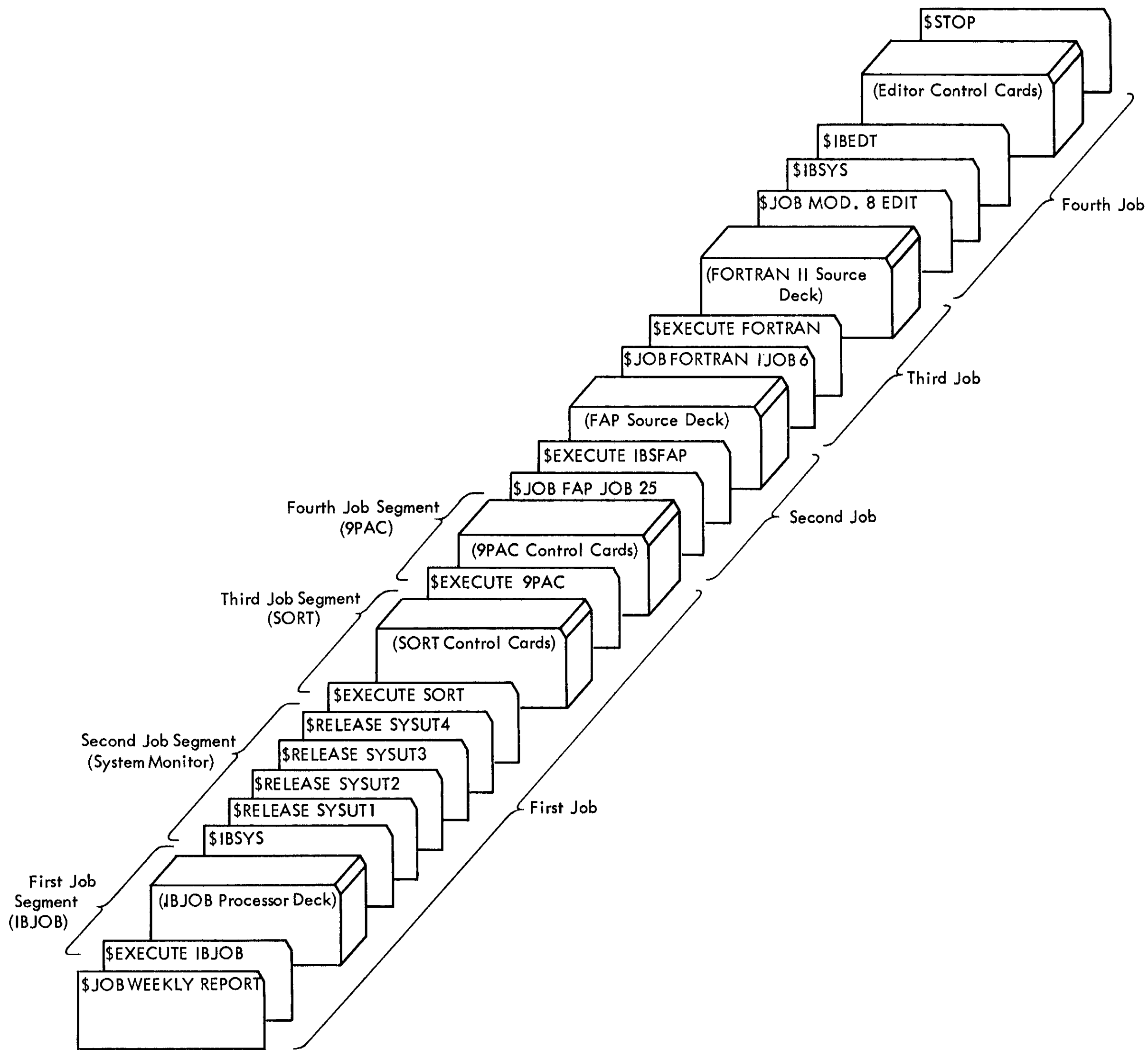


Figure 3. Composite System Input Deck

play or otherwise use the SYSDAT word, it should not be modified by the subsystem. If an installation has an interval timer that produces the current date, the date should be stored in the form specified for the $DATE card, where:

mm = Month (01 to 12)
dd = Day (01 to 31)
yy = Year (00 to 99)

$RESET Card

Format:

```
1             16
$RESET
```

This card causes the assignment of a unit to any system unit function, which currently has no unit assigned, but had a unit assigned at initial start. The original unit is reassigned to the function if the unit is not reserved or detached. If it is reserved or detached, a unit from a unit availability chain in the System Nucleus is assigned.

The $RESET card may be used following a $IBSYS card and preceding a $EXECUTE card to ensure that all system utility functions have units assigned before the start of a new job segment.

$RESTORE Card

Format:

```
1             16
$RESTORE
```

This card causes the restoration of the System Supervisor and the regeneration of the System Nucleus as specified by assembly parameters.

The $RESTORE card causes the System Monitor to be called into core storage from the System Library Unit, giving the same effect as an initial start, except that the tape positions and the SYSDAT word in the Nucleus are not disturbed. The effect of all previous control cards is canceled, except that the $RESTORE card does not effect the source of input as specified by $CARDS or $TAPE cards. However, the unit assigned as the System Card Reader or the System Input Unit may change as a result of the $RESTORE card because of a different unit having been assigned previously by $ATTACH, $AS, or $SWITCH cards.

$RESTART Card

Format:

```
1              16
$RESTART    [{±n    }]
             {MATCH }
```

This card is used in the System Card Reader to restart at the beginning of a particular job on the System Input Tape. It may be used by the operator in performing an initial start procedure, a between-jobs interrupt procedure, an end-of-jobs procedure, or a procedure following the detection of an invalid System Monitor control card. The exact use of the $RESTART card in performing each of these procedures is described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

If the variable field of the $RESTART card is +n, the System Supervisor will restart at the beginning of the nth job following the last completed or last partially completed job. For example, if n is 1 and the card is used during an interruption between jobs on the System Input Tape, the System Supervisor restarts at the beginning of the next job on the input tape as though no $RESTART card was processed. If the variable field is −n, the System Supervisor will restart at the nth job preceding the last completed or partially completed job. The value n may range from 0 to 9999. If n is 0 or blank, the System Supervisor will restart at the beginning of the last completed or partially completed job.

When the word MATCH is specified in the variable field of the $RESTART card, the card should be followed by a $JOB card corresponding to a $JOB card on the System Input Tape. The System Supervisor will read the $JOB card following the $RESTART MATCH card, rewind the System Input Tape, search the input file for a job with a matching $JOB card and, if found, restart at the beginning of the job. If a $STOP card is encountered before a matching job card, the System Input Tape will be repositioned to the end of the last completed job and the restart request will be ignored.

NOTE: Only nonblank characters in the variable field will be compared when matching job cards.

Miscellaneous Control Cards

$ID Card

Format:

```
1              7
$ID            any text
```

This card is used for intrajob accounting purposes at installations that employ an installation accounting routine. The card causes a transfer of control to the installation accounting routine if one exists. Upon completion of the accounting routine, the next card in the system input file is read and processed. Columns 7 through 72 of the card may contain any combination of alphameric characters and blanks.

The distributed version of the Operating System does not contain an installation accounting routine. Therefore, no action occurs when the card is read other than the listing of the card on the System Printer and the System Output Unit.

The exact use and placement of the $ID card will depend upon the accounting practices and regulations in force at a particular installation.

$* Card

Format:

```
1              3
$*             any text
```

This card is listed on the System Printer and System Output Unit. No further action occurs. Columns 3 through 72 of the card may contain any combination of alphameric characters and blanks.

$PAUSE Card

Format:

```
1              16
$PAUSE         instructions to operator
```

This card causes the machine to stop after the card and the following message is listed on the System Printer:

```
OPER.    ACTION    PAUSE
```

When START is pressed, the processing of cards on the system input file is resumed. Columns 16 through 72 of the card may contain any combination of alphameric characters and blanks.

This card provides the programmer with a means of temporarily interrupting processing to enable the operator to perform a specific task, such as file protecting a reel of tape. Therefore, when the card is used, it should contain an explicit message to the operator so that processing can continue without further delay.

$LIST Card

Format:

```
1              16
$LIST
```

This card causes all control cards to be listed on the System Printer as well as on the System Output Unit. Normally, all cards are listed on the System Output Unit and only the $JOB, $ID, $*, $PAUSE, $RESTART, $STOP, $CARDS, $TAPE, $LIST, and $UNLIST cards are listed on the System Printer.

$UNLIST Card

Format:

```
1              16
$UNLIST
```

This card negates the effect of the $LIST card by causing only the $JOB, $ID, $*, $PAUSE, $RESTART, $STOP, $CARDS, $TAPE, $LIST, and $UNLIST cards to be listed on the System Printer. The normal mode is UNLIST, unless the control cards are read by the System Card Reader.

$UNITS Card

Format:

```
1              16
$UNITS
```

This card causes all systems unit function names, physical unit assignments, and assigned densities to be listed on the System Output Unit. If a direct access storage unit is assigned to a system unit function, the HA2 home address identifier and the cylinder limits for the function are included.

This information is also printed on the System Printer if the $UNITS card was read from the System Card Reader or if a $LIST card is in effect.

The printout caused by the $UNITS card may be used to verify all unit assignment operations.

$IBEDT Card

Format:

```
1              16
$IBEDT
```

Upon recognizing this card, the System Supervisor calls the System Editor into core storage from a System Library Unit and relinquishes control to it. The same effect may be accomplished by a $EXECUTE card with the name EDITOR specified in the variable field. The control cards that are recognized by the System Editor are described in the section "System Editor."

Unit Assignment Control Cards

The purpose of the unit assignment control cards is twofold. First, they provide a means whereby an installation may indicate changes in machine input/output capabilities to the System Monitor and the subsystems under its control. Second, they provide a means for changing input/output unit assignments within a job.

The unit assignment control cards fall into two categories: those which define the physical avaliability (attachment or detachment) of an input/output unit and those which reassign input/output units to logical system unit functions. Input/output units are initially assigned by assembly parameters when the IBSYS Operating System is assembled. The unit assignment control cards are normally used only for the temporary reassignment of units.

Unit Designation

Physical input/output units and logical system unit functions are designated on the unit assignment control cards as described below.

729 Magnetic Tape Units: A 729 tape unit is designated as xk, where x is the channel (A through H) and k is the tape unit number (0 through 9) on that channel.

Card and Printer Units: The card and printer units are designated as RDx, PUx, and PRx, where RD, PU, and PR are the card reader, card punch, and printer, respectively, and x is the channel (A through H).

Disk Storage Units: A disk storage unit is designated as xDam/s, where x is the channel (A through H), D designates disk, a is the access arm (0 or 1), m is the module (0 through 9), and s is the setting of the Data Channel Switch (0 for switch setting 1; 1 for switch setting 2). If the /s is missing from the unit designation, switch setting 1 is assumed.

Drum Storage Units: A drum storage unit is designated as xNam/s, where x is the channel (A through H), N designates drum, a is the access arm (0), m is the module (0, 2, 4, 6, or 8), and s is the setting of the Data Channel Switch (0 for switch setting 1; 1 for switch setting 2). If the /s is missing from the unit designation, switch setting 1 is assumed.

7340 Hypertape Drives: A 7340 Hypertape Drive is designated as xHk/s, where x is the channel (A through H), H designates Hypertape, k is the drive number (0 through 9), and s is the setting of the Data Channel Switch (0 for switch setting 1; 1 for switch setting 2). If the /s is missing from the unit designation, switch setting 1 is assumed.

System Unit Functions: A system unit function is designated as SYSxxx or SYSyyy, where SYSxxx or SYSyyy is the symbolic name for one of the system unit functions listed in Figure 4.

$DETACH Card

Format:

```
1              16
$DETACH        Unit
```

This card causes the specified unit to be detached. If the unit is assigned to a system unit function, the assignment is cancelled. The unit will remain un-

Symbolic Name	Function
SYSLB1	Library 1
SYSLB2	Library 2
SYSLB3	Library 3
SYSLB4	Library 4
SYSCRD	Card Reader
SYSPRT	Printer
SYSPCH	Card Punch
SYSOU1	Output
SYSOU2	Alternate Output
SYSIN1	Input
SYSIN2	Alternate Input
SYSPP1	Peripheral Punch
SYSPP2	Alternate Peripheral Punch
SYSCK1	Checkpoint
SYSCK2	Alternate Checkpoint
SYSUT1	Utility 1
SYSUT2	Utility 2
SYSUT3	Utility 3
SYSUT4	Utility 4
SYSUT5	Utility 5
SYSUT6	Utility 6
SYSUT7	Utility 7
SYSUT8	Utility 8
SYSUT9	Utility 9

Figure 4. Symbolic Names of System Unit Functions

available and unassigned until it is made available by a $ATTACH card, or until its status is restored by a $RESTORE card or an initial start. The $DETACH card may be used to temporarily deactivate a unit for maintenance purposes.

$ATTACH Card

Format:

```
1             16
$ATTACH       unit [,II]
```

This card causes the specified unit to become attached. The unit attached by this card can be assigned to a system unit function by the next $AS card.

If the specified unit is a Model II or Model V 729 Magnetic Tape Unit, this must be specified by II in the variable field of the card. If II does not appear in the variable field when a 729 Magnetic Tape Unit is specified, the System Supervisor assumes that the unit is a Model IV or VI.

The $ATTACH and $DETACH cards may be used to alert the System Monitor to a physical change in the status of a unit. For example, if, at an installation, the sixth unit on channel D were physically disconnected and then reconnected as the fifth unit on channel C, this change in status might be indicated to the System Monitor by the following cards:

```
1             16
$ATTACH       C5
$DETACH       D6
```

$AS Card

Format:

```
1             16
$AS           SYSxxx [,{H}]
                       {L}
```

This card causes the unit specified on the last recognized $ATTACH card to be assigned to the specified system unit function. If the tape density is specified by H, HI, or HIGH, the density for the system unit function is set to high. It is set to low if the specifications is L, LO, or LOW. If the density specification is absent, the density will be set according to the assembly parameter HIGHLO. With the distributed System Library Tape, high density is assumed if the density is not specified on the card.

This card may apply to a unit already attached. In this case, there is no need to detach the unit before reattaching it. The $AS card releases the unit that was assigned to the system unit function before the $AS card was recognized. If there are two or more $AS cards in succession, the unit specified on the last recognized $ATTACH card is assigned to the function specified on each $AS card.

If the last attached unit was a direct access storage unit, the format of the $AS card is:

```
1             16
$AS           SYSxx,nnn,ccc,hh
```

Here, nnn and ccc must be replaced by three-digit numbers denoting the number of cylinders (nnn) and the starting cylinder (ccc) of the direct access storage unit assigned to the system unit function. A two-character symbol, representing the HA2 home address identifier, should be placed in the next field (hh). Any blanks or zeros in this field will be converted to octal 12s.

For example, to assign 25 cylinders of a 1301 or 2302 Disk Storage Unit [starting with cylinder 125 (load point) on access arm 0, in module 0 on channel E, switch setting 2, with a HA2 of PQ] as System Utility Unit 3 (SYSUT3), the following control cards would be used:

```
1             16
$ATTACH       ED00/1
$AS           SYSUT3,025,125,PQ
```

Note that the/1 corresponds to a switch setting of 2.

$RELEASE Card

Format:

```
1             16
$RELEASE      SYSxxx
```

This card causes the unit assigned to the specified system unit function to be released from the function. If the unit was concurrently assigned to other system unit functions, it remains assigned to those functions.

$SWITCH Card

Format:

```
1             16
$SWITCH       SYSxxx,SYSyyy
```

This card causes the units assigned to the two specified system unit functions to be transposed; that is, the unit assigned to SYSxxx is assigned to SYSyyy, and

the unit that was assigned to SYSyyy is assigned to SYSxxx. Physical density settings of the units remain the same.

$CARDS Card

Format:

1 16
$CARDS

This card causes the System Supervisor to read succeeding control cards from the unit assigned as the System Card Reader (SYSCRD).

$TAPE Card

Format:

1 16
$TAPE

This card causes the System Supervisor to read succeeding control cards from the unit assigned as the System Input Unit (SYSIN1).

Tape Manipulation Control Cards

The tape manipulation control cards provide the operator and programmer with facilities for the automatic manipulating of tape units assigned to system unit functions. If no unit is assigned to the system unit function specified on a tape manipulation control card, or if the unit assigned to the function is not a 729 tape unit or a 7340 Hypertape Drive, the card has no effect. The $ENDFILE, $REWIND, and $REMOVE cards have the same effect on a 7340 Hypertape Drive and a 729 Magnetic Tape Unit.

$ENDFILE Card

Format:

1 16
$ENDFILE SYSxxx

This card causes the tape unit assigned to the specified system unit functions to write, on the tape, an end-of-file gap followed by a tape mark. No test is made to determine if the operation is invalid, such as writing a tape mark on the System Input Unit.

$REWIND Card

Format:

1 16
$REWIND SYSxxx

This card causes a tape unit assigned to the specified system unit function to be rewound. If the specified function is SYSOU1, further use of the System Output Unit by the System Supervisor is suspended until a $JOB card is read or the System Supervisor is called into core storage again by a $IBSYS card or by a subsystem.

$REMOVE Card

Format:

1 16
$REMOVE SYSxxx

This card causes a tape unit assigned to the specified system unit function to be rewound and unloaded. If the specified function is SYSOU1, further use of the System Output Unit by the System Supervisor is suspended until a $JOB card is read or the System Supervisor is called into core storage again by a $IBSYS card or by a subsystem.

$UNLOAD Card

Format:

1 16
$UNLOAD SYSxxx

This card causes the Hypertape assigned to the specified system unit function to be unloaded without rewinding. If a 729 tape unit is assigned to the specified system unit function, the $UNLOAD card will be interpreted as a $REMOVE card.

$PROTECT Card

Format:

1 16
$PROTECT SYSxxx

This card causes the Hypertape assigned to the specified system unit function to be file-protected. The $PROTECT card is ignored if a Hypertape drive is not assigned to the specified system unit function.

System Core-Storage Dump Program

General Description

The System Core-Storage Dump Program is designed to facilitate the testing of programs under System Monitor control. To perform this function, snap dump and post-mortem dump options have been provided. The snap dump option of the Core-Storage Dump Program can dump and edit one or more sequential locations of core storage during the execution of an object program. After the dump is completed, core storage is returned to its original condition and control is returned to that point in the object program from which the dump was called. The post-mortem dump option of the Core-Storage Dump Program dumps core storage in the same manner as the snap dump option, but at its completion control is returned to the System Supervisor which then skips cards on the System Input File until a $STOP card or the next job or job segment is encountered. It then begins to process control cards. A job always begins with a $JOB card, and a job segment always begins with a $EXECUTE or $IBSYS card.

The format, the limits, and the output units for both the snap and post-mortem forms of the core storage dump may be either assembly defined or specified by a control word*. For the post-mortem dump, the additional option of defining these parameters with the console entry keys is provided.

When a dump is requested, the System Nucleus writes a portion of core storage onto the alternate System Peripheral Punch Unit (SYSPP2), reads in the Core-Storage Dump Program, and transfers control to it. If the console entry key option is requested, a halt occurs to allow the insertion of dump parameters. The Core-Storage Dump Program dumps the edited output onto the System Output Unit, onto the System Printer, or onto both.

Using the System Core-Storage Dump Program

Transfer to Dump Instructions

To obtain a dump of core storage during the execution of an object program, insert one of the following instructions in the body of the source program at the point at which the dump is required:

TRA SYSDMP: This instruction provides a post-mortem dump of all of core storage in an assembly defined format.

TRA SYSDMP with Sense Switch 4 Down: This instruction results in a post-mortem dump, in accordance with the parameters entered by means of the console entry keys. Before the dump is executed, a halt occurs and a message requesting insertion of dump parameters is printed on the System Printer.

TSX SYSDMP, 4, 1 Followed by a Parameter Control Word: This sequence results in a post-mortem dump, in accordance with the information from the parameter control word. The parameter control word is described in the section "Dump Parameters."

TSX SYSDMP, 4, 0 Followed by a Parameter Control Word: This sequence results in a snap dump, in accordance with the information from the parameter control word. For example, the following instruction and control word would result in an octal snap dump on the System Output Tape of storage locations beginning at STDMP and ending at ENDMP.

```
8       16
TSX     SYSDMP,4,0
PON     ENDMP,0,STDMP
```

Dump Parameters

The dump parameters are entered either by a parameter control word (Figure 5) for a snap dump or by a parameter control word or the console entry keys for a post-mortem dump. Any one of six dump formats (Figure 6) can be specified. In the distributed version of the Core-Storage Dump Program, output is single spaced. The various parts of the parameter control word are interpreted as follows:

Prefix	PON	1 FORMAT A – Octal, eight words per line.
	PTW	2 FORMAT B – BCD, sixteen words per line.
	PTH	3 FORMAT C – SQUEZY, Mnemonics with address and tag field. If the Core-Storage Dump Program cannot interpret the operation code, the octal representation is given.
	MZE	4 FORMAT D – Octal and SQUEZY. If the SQUEZY word would normally have appeared in octal form, it is not listed twice, but is suppressed. Otherwise, both the octal word and the SQUEZY word are listed.

*If no parameters are specified when a dump is requested, the dump will be accomplished using parameters within the Core-Storage Dump Program. These parameters are established when the Core-Storage Dump Program is assembled. (See the section "System Library Preparation and Maintenance.")

	MON	5 FORMAT E – Octal and mnemonics.
	MTW	6 FORMAT F – Octal, mnemonics, and BCD. A BCD interpretation of the word is listed to the right of the mnemonic.
Address	The ending location of the dump.	
Tag	0 = Dump onto System Output Unit. 1 = Dump onto System Printer. 2 = Dump onto both System Output Unit and System Printer.	
Decrement	The starting location of the dump.	

Prefix	Decrement	Tag	Address
Format Code	Starting Location	Output Code	Ending Location
S——2	3——17	18——20	21——35

Figure 5. Parameter Control Word Format

NOTES: The limits of the requested dump may be stated in any order, i.e., the starting parameter in the address field and the ending parameter in the decrement field, or vice versa.

In the distributed version of the Core-Storage Dump Program, the dump is always made onto the System Output Unit only, regardless of the contents of the tag position of the parameter control word.

A parameter control word of all zeros will provide a full core storage dump in the assembly defined format. In the distributed version, format 3(C) in Figure 6 is used.

Machine Status at the End of a Core Storage Dump

If any input/output operation had been in progress when the snap dump routine was called, the traps resulting from this operation are lost if they occurred on the channels used by the Core-Storage Dump Program. Since the System Loader disables all traps when loading the Core-Storage Dump Program, and the dump resets the work tape data channel when restoring core storage, it is recommended that all input/output operations be terminated before calling in the Core-Storage Dump Program.

At the completion of a snap dump, all of core storage, except locations SYSEND-18 through SYSEND, are restored. These last 19 locations are destroyed by the restore portion of the snap dump routine. Because of space restrictions, no error checking of the work tape is done while restoring core storage. The snap dump routine also repositions the System Library Tape to its position prior to the core storage dump as part of its restore process.

Format A -- Octal

```
     AC             MQ          SENSE IND       KEYS           XR1      XR2      XR4
000000000000  000000000000  001321000000  000000000000     00001    00006    74320
                                                           -77777   -77772   -03460
          INDICATORS                             SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL             1      2      3      4             1      2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF    ON            OFF    OFF    OFF    OFF            ON     OFF    OFF    ON     OFF    ON
00000  100000000000  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
00030  000000000031  000060000030  000000000033  000060000032  001321000000  000060000034  001321000000  001321000000
```

Format B -- BCD

```
     AC             MQ          SENSE IND       KEYS           XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000     00001    77323    71371
                                                           -77777   -00455   -06407
          INDICATORS                             SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL             1      2      3      4             1      2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF    ON            OFF    OFF    OFF    OFF            ON     OFF    OFF    ON     OFF    ON
00000 DK90UT 800000 4+003* 0+00** 0A00*F 000000 000000 0A00*H 0A00*+ 000000 000000 0A007R 000000 0A007R 000000 0A007R
00020 000000 0A007R 000000 0A007R 00000E 00 00D 00000G 00 00F 00000I 00 00H 00000. 00 00+ 801H4* 00 00) 801H4* 801H4*
00040 000000 000000 000000 80008U 000000 80108U 000000 80208U 000000 80308U 000000 80408U 000000 80508U 000000 80608U
```

Format C -- SQUEZY

```
     AC             MQ          SENSE IND       KEYS           XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000700     00001    77323    71371
                                                           -77777   -00455   -06407
          INDICATORS                             SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL             1      2      3      4             1      2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF    ON            OFF    OFF    OFF    OFF            ON     OFF    OFF    ON     OFF    ON
00000  244211006463  100000000000  HPR      377  TRA     5557  TTR     3626  HTR           HTR           TTR     3630
00010  TTR     3632  HTR           HTR           TTR      751  HTR           TTR      751  HTR           TTR      751
00020  HTR           TTR      751  HTR           TTR      751  HTR       25  HTR *     24  HTR       27  HTR *     26
```

Format D -- Octal and SQUEZY

```
     AC             MQ          SENSE IND       KEYS           XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000     00001    77323    71371
                                                           -77777   -00455   -06407
          INDICATORS                             SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL             1      2      3      4             1      2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF    ON            OFF    OFF    OFF    OFF            ON     OFF    OFF    ON     OFF    ON
00000  244211006463  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
                                   HPR      377  TRA     5557  TTR     3626  HTR           HTR           TTR     3630
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
       TTR     3632  HTR           HTR           TTR      751  HTR           TTR      751  HTR           TTR      751
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
       HTR           TTR      751  HTR           TTR      751  HTR       25  HTR *     24  HTR       27  HTR *     26
```

Format E -- Octal and Mnemonics

```
     AC             MQ          SENSE IND       KEYS           XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000     00001    77323    71371
                                                           -77777   -00455   -06407
          INDICATORS                             SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL             1      2      3      4             1      2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF    ON            OFF    OFF    OFF    OFF            ON     OFF    OFF    ON     OFF    ON
00000  244211006463  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
       TIX           TXI           HPR           TRA           TTR           HTR           HTR           TTR
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
       TTR           HTR           HTR           TTR           HTR           TTR           HTR           TTR
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
       HTR           TTR           HTR           TTR           HTR           HTR *         HTR           HTR *
00030  000000000031  000060000030  000000000033  000060000032  101201300475  000060000034  101201300475  101201300475
       HTR           HTR *         HTR           HTR *         TXI           HTR *         TXI           TXI
```

Format F -- Octal, Mnemonics, and BCD

```
     AC             MQ          SENSE IND       KEYS           XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000     00001    77323    71371
                                                           -77777   -00455   -06407
          INDICATORS                             SENSE LIGHTS             LOC            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL             1      2      3      4   02160       1      2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF    ON            OFF    OFF    OFF    OFF            ON     OFF    OFF    ON     OFF    ON
00000  244211006463  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
       TIX   DK90UT  TXI   800000  HPR   4+003*  TRA   0+00**  TTR   0A00*F  HTR   000000  HTR   000000  TTR   0A00*H
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
       TTR   0A00*+  HTR   000000  HTR   000000  TTR   0A007R  HTR   000000  TTR   0A007R  HTR   000000  TTR   0A007R
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
       HTR   000000  TTR   0A007R  HTR   000000  TTR   0A007R  HTR   00000E  HTR * 00 00D  HTR   00000G  HTR * 00 00F
```

Figure 6. Core-Storage Dump Formats

Function and Organization

The System Nucleus is a portion of the System Monitor which remains in core storage at all times and provides common facilities for intercommunication and control between the System Monitor and the subsystems, and between the subsystems themselves. The Input/Output Executor and the unit control blocks for the input/output units are described separately in the section "Input/Output Executor" although they may be considered part of the System Nucleus in that they normally remain in core storage at all times and provide common facilities for the System Monitor and the subsystems.

The System Nucleus consists of the following:

1. The Communication Region containing constants, control words, and transfer words that are required for intercommunication between the System Monitor and the subsystems.

2. The System Unit Function (SYSUNI) Table, which is used to keep account of the units assigned to system unit functions.

3. The Disk/Drum Limits Table, which when direct access storage is employed, supplements the function of the System Unit Function Table by defining the portion of disk or drum storage assigned to each system unit function.

4. The Unit Control Block Table, which consists of a word for each channel containing the address of the first unit control block for the channel, the total number of input/output units assigned to the channel, and the number of card units assigned to the channel.*

5. The Unit Availability Table, which consists of one word for each channel that serves as an entry point to a chain of available units on the channel.

6. The System Loader, which is used by the System Monitor and the subsystems for scatter-loading records from the System Library Unit.

7. An interrupt routine, which is used by the System Monitor and the subsystems to interrupt processing between jobs for the purpose of manual intervention.

8. A dump-calling routine, which is a bootstrap routine for loading the Core-Storage Dump Program whenever a dump is called for by the System Monitor, a subsystem, an object program, or manually by the operator.

Of the various portions of the System Nucleus, the most important, so far as the systems programmer is concerned, are the Communication Region, the System Unit Function Table, the Disk/Drum Limits Table, and the Unit Availability Table. Therefore, each of these is described below, together with job control communication requirements for subsystems operating under the System Monitor.

*Each input/output unit that may be referred to by the IBSYS Operating System is represented internally by a block of four consecutive words called a unit control block. The unit control block contains information concerning the status of each input/output device. A complete description of unit control blocks is provided in the section "Input/Output Executor."

Communication Region

The Communication Region of the System Nucleus consists of 32 consecutive core storage locations containing various constants, control words, and transfer words that may be referred to by the System Monitor or by any of the subsystems operating under it. The function of each entry in the Communication Region, together with its octal absolute address and its symbolic address, is given in Figure 7. A more complete description of the function of each entry is given in Appendix A.

The entries in the Communication Region may be referred to by their absolute addresses, since their

Octal Address	Symbolic Address	Function
100	SYSTRA	Transfer instruction to current subsystem
101	SYSDAT	Date Word
102	SYSCUR	Name of current subsystem
103	SYSRET	Location to which each subsystem returns
104	SYSKEY	Contents of entry keys at initial start
105	SYSSWS	Contents of sense switches at initial start
106	SYSPOS	Initial position and index of current subsystem
107	SYSUNI	Location and length of System Unit Function Table
110	SYSUBC	Location and length of table of unit control block locators by channel
111	SYSUAV	Location and length of the Unit Availability Table
112	SYSUCW	Location and length of all the unit control blocks
113	SYSRPT	Transfer to between-jobs interrupt routine
114	SYSCEM	Transfer to customer engineering diagnostic routine
115	SYSDMP	Transfer to bootstrap for core storage dump
116	SYSIOX	Location and length of IOEX communication Table
117	SYSIDR	Transfer to installation accounting routine, if any.
120	SYSCOR	Lower limit of usable core storage in decrement, upper limit in address
121	SYSLDR	Transfer to system scatter-load routine
122	SYSACC	Installation accounting routine Communication
123	SYSPID	Installation accounting routine Communication
124	SYSCYD	Channel commands for system to copy and disconnect
125	SYSCYD+1	
126	SYSSLD	Self-loading sequence
127	SYSTCH	Self-loading sequence
130	SYSTCH+1	
131	SYSTWT	System trap, wait, and transfer point
132	SYSGET	Subsystem communication with System Supervisor
133	SYSJOB	Job control word
134	.CHEXI	Cells tested by all subsystems to determine whether in direct-couple environment or not
135	.MODSW	
136		Reserved for future IBM use
137		Reserved for future IBM use

Figure 7. Communication Region of System Nucleus

location will not change and is not dependent on the input/output configuration of the Operating System. In addition, these entries may be referred to by their symbolic addresses (without further definition in the source program) in any Macro Assembly Program (MAP) relocatable assembly, with or without MONSYM specified; in a MAP absolute assembly where the MONSYM or JOBSYM option is specified on the $IBMAP control card; or in a FORTRAN II Assembly Program (FAP) assembly if a Save Symbol Table (SST) pseudo-operation is included in the first card group of the FAP source program.

System Unit Function Table

The System Unit Function Table (SYSUNI) consists of 24 entries, one for each of the 24 possible system unit functions. The octal absolute address, the symbol, and the system unit function for each entry are given in Figure 8. The entries in this table may be referred to by their symbols (without further definition in the source program) in any MAP relocatable assembly, with or without MONSYM specified; in a MAP absolute assembly where the MONSYM or JOBSYM option is specified on the $IBMAP control card; and in a FAP assembly where a Save Symbol Table (SST) pseudo-operation is included in the first card group of the FAP language source program. In the last two cases, the symbols listed in Figure 8 become the symbolic addresses of the table entries, that is, the symbols defined by the assemblers are equal to their octal addresses.

Octal Address	Symbol	System Unit Function
140	SYSLB1	Library 1
141	SYSLB2	Library 2
142	SYSLB3	Library 3
143	SYSLB4	Library 4
144	SYSCRD	System Card Reader
145	SYSPRT	System Printer
146	SYSPCH	System Punch
147	SYSOU1	Output
150	SYSOU2	Alternate Output
151	SYSIN1	Input
152	SYSIN2	Alternate Input
153	SYSPP1	Peripheral Punch
154	SYSPP2	Alternate Peripheral Punch
155	SYSCK1	Checkpoint 1
156	SYSCK2	Checkpoint 2
157	SYSUT1	Utility 1
160	SYSUT2	Utility 2
161	SYSUT3	Utility 3
162	SYSUT4	Utility 4
163	SYSUT5	Utility 5
164	SYSUT6	Utility 6
165	SYSUT7	Utility 7
166	SYSUT8	Utility 8
167	SYSUT9	Utility 9

Figure 8. System Unit Functions

For example, the symbol SYSLB1 represents 140, and SYSUT5 represents 163. However, in a MAP relocatable assembly, if the MONSYM option is not specified on the $IBMAP control card, the symbols defined at load time are equal to index numbers beginning with 0 for SYSLB1 and ending with 27_8 for SYSUT9. In order to obtain the correct address of an entry in the SYSUNI table when the symbols are defined in this way, the index number represented by the symbol for that entry must be added to the contents of the address portion of the SYSUNI entry in the Communication Region of the Nucleus. Since the address portion of this word contains the address of the first word of the System Unit Function Table, the resulting sum is equal to the address of the SYSUNI table entry represented by the symbol used in the addition. For example, adding SYSOU1 (defined as 7) to the address portion of the SYSUNI entry (140_8) results in the address (147_8) of the entry in the SYSUNI table for the System Output Unit.

The address portion of each entry in the System Unit Function Table contains the address of the first word of the unit control block for the unit assigned to the system unit function. If no unit is assigned to the system unit function, the address portion of the entry contains zeros. Normal assignments of units to system unit functions are specified when the System Monitor is assembled. However, normal unit assignments may be changed temporarily by unit assignment control cards.

Tape density is indicated by an entry in the sign bit position. Low density is specified when the sign is plus (PZE), and high density is specified when the sign is minus (MZE). The System Monitor will only set the densities of the tape units it uses. Each subsystem has the responsibility for setting the densities of the tape units it uses.

When direct access storage is assigned to a system unit function, bits 6 through 17 of the entry for the system unit function contain the HA2 home address identifier that was specified on the $AS control card used to assign the disk or drum module to the function. Bits 3 through 5 are zero.

HA2 Table

An auxiliary system unit function table, the Home Address 2 (HA2) Table, is located in the System Supervisor and has 24 entries corresponding to those of the System Unit Function Table. The HA2 Table, which defines the HA2 of each system unit function, is initially assembled with BM for the HA2's of SYSLB1 through SYSLB4, and 00 (OCT 1212) for the HA2's of the remaining System Unit functions (excluding SYSCRD, SYSPRT, and SYSPCH).

Although this table is not located in the Nucleus, HA2's are generated into the decrement of the System Unit Function Table entries from this table for those system units which are permanently assigned to disk or drum. The symbolic location of this table is HA2TBL.

Disk/Drum Limits Table

The Disk/Drum Limits Table is located immediately following the System Unit Function Table. It contains 24 entries corresponding to the 24 entries of the System Unit Function Table. When an entry in the System Unit Function Table contains the address of a unit control block for a direct access storage module, the corresponding entry in the Disk/Drum Limits Table contains a parameter word, in binary form, which designates the first and last tracks of the consecutive tracks of the module that are assigned to the system unit function. Since direct access storage cannot be assigned to the System Card Reader, System Printer, and System Card Punch functions, the entries in the Disk/Drum Limits Table corresponding to the entries for these three functions in the System Unit Function Table are used for other purposes by the operating system. The parameter words in the Disk/Drum Limits Table have the following format:

```
PZE     DORG,,DEND
```

Unit Availability Table

The Unit Availability Table consists of one entry per input/output channel, beginning with channel A. Each entry contains the address of the first word of the unit control block for the first unassigned (available) unit on the channel. The address portion of the first word of the unit control block for each unassigned unit on a channel contains the address of the first word of the unit control block for the next unassigned unit on the channel. In this way, a unit availability chain for each channel is formed, beginning with an entry in the Unit Availability Table. The end of the chain is indicated by zeros in the address portion of the first word of the unit control block for the last unassigned unit on the channel. Whenever a subsystem requires the use of an available unit, it interrogates a unit availability chain by way of a Unit Availability Table entry and removes the unit from the chain.

Job Control Communication with Subsystems

Each of the subsystems operating under the System Monitor must follow certain operation procedures involving communication with the System Monitor. These procedures are required to ensure proper job and job segment control, proper control of between-jobs and end-of-jobs interrupts, and restoration, if necessary, of input/output unit assignments at the beginning of each job.

Restoration of Unit Assignments Between Jobs

Restoration of input/output unit assignments is required at the beginning of a job if, during the previous job, a $IBSYS card was followed by a $AS, $SWITCH, or $RELEASE card or a subsystem (or an object program running under the subsystem) removed or replaced a unit in a unit availability chain. If either of these conditions occurred, the System Supervisor, at the beginning of the next job, begins restoring unit assignments by first chaining all attached units on each channel in ascending order in accordance with the unit or module number. Any unit assigned to a system unit function at initial start is then reassigned by the System Supervisor to the same function, provided the unit is not currently assigned to a system input, system output, or system peripheral punch function or is not currently assigned to a system unit function in place of a detached unit. Any unit assigned to a system unit function is removed from the unit availability chain for its channel unless it is a direct access storage module. The restoration of unit assignments at the beginning of a job differs from that initiated by the $RESTORE card in that detached units remain detached and current system input, system output and system peripheral punch assignments are not changed.

Communication Region Locations SYSRET, SYSGET and SYSJOB

Communications between the subsystems and the System Supervisor is carried out chiefly by way of three locations in the Communication Region of the System Nucleus. These three locations are SYSRET, SYSGET and SYSJOB. The function of each of these locations is described below.

Communication Location SYSRET

Whenever a subsystem is required to return control to the System Supervisor, it transfers to the location SYSRET. As a result, the System Supervisor is read into core storage and control is relinquished to it.

Communication Location SYSGET

Before returning control to the System Supervisor, a subsystem must ensure that a word is stored in the SYSGET location which indicates to the System Supervisor the reason why control was returned to it by the subsystem. In addition, whenever a post-mortem dump is performed by the System Core-Storage Dump Pro-

gram it places a word in SYSGET (to indicate that a post-mortem dump was performed) before it returns control to the System Supervisor. When the System Supervisor obtains control from the dump program or from a subsystem, it examines the word in SYSGET and takes appropriate action.

If the word in the SYSGET location is a subsystem name, it indicates to the System Supervisor that a subsystem read a $EXECUTE card containing a subsystem name other than its own. Therefore, the System Supervisor loads into core storage the first record of the subsystem whose name was in the SYSGET location and relinquishes control to it.

If the word in the SYSGET location is "IBSYST," it indicates to the System Supervisor that a $IBSYS control card was read by a subsystem. Therefore, the System Supervisor begins processing control cards on the input file, beginning with the next card.

If the word in the SYSGET location is "IBSXEC," it indicates to the System Supervisor either of the following:

1. A post-mortem core-storage dump was taken by a subsystem, an object program, or the operator; therefore, a job segment was not completed.
2. A subsystem could not complete a job segment.

In either case, the System Supervisor skips on the system input file until a $IBSYS, $EXECUTE, $JOB, or $STOP control card is encountered and then processes cards, normally beginning with that card.

If the word in the SYSGET location is "IBSNXT," it indicates to the System Supervisor that a subsystem has determined that a job cannot be completed. Therefore, the System Supervisor skips on the system input file until a $JOB or $STOP control card is encountered and then processes cards normally, beginning with that card.

If the word in the SYSGET location is "$STOPb," where "b" is a blank, it indicates to the System Supervisor that a $STOP control card was read by a subsystem. Therefore, the System Supervisor initiates an end-of-jobs sequence as though it had read the $STOP card.

If the word in the SYSGET location is "IBSBSR," it indicates to the System Supervisor that a $JOB card was read by a subsystem and either a between-jobs interrupt condition exists or an input/output unit was reassigned or made unavailable during the previous job. In either case, the System Supervisor proceeds as though it had just read a $JOB card. Refer to the description of the $JOB card in the section "System Supervisor."

Communication Location SYSJOB

The communication location SYSJOB is used by the System Supervisor and the subsystems in controlling the processing of jobs.

The sign bit of the word indicates whether or not any input/output unit assignments need be restored at the beginning of the next job. A minus sign indicates that restoration is necessary and a plus sign indicates that restoration is not necessary. The System Supervisor sets the sign of SYSJOB to minus if it processes a $AS, $SWITCH, or $RELEASE control card. Similarly, each subsystem must set the sign to minus if it is about to change a unit availability chain. The sign should be set to minus before the change is made in the event the job "blows-up" and the subsystem does not regain control. The sign bit is interrogated at the beginning of each new job by the Nucleus routine SYSRPT to determine if restoration of input/output unit assignments is required. It is then set to plus by the System Supervisor before actual processing of the job begins.

Bit 17 of SYSJOB is used by the subsystems to indicate to one another whether or not a previous job segment could not be completed. Whenever a subsystem (or Editor) determines that it cannot complete a job segment, it sets bit 17 of SYSJOB to a 1 and stores the word "IBSXEC" in the SYSGET location. It then returns control to the System Supervisor, which skips to the next job segment.

When either a subsystem or the System Editor gains control at the beginning of a job segment, it tests bit 17 of SYSJOB and proceeds normally if it is a 0. However, if it is a 1, the subsystem may discontinue the present job segment by storing the word "IBSXEC" in the SYSGET location and returning control to the System Supervisor by way of SYSRET. This is done in cases where the successful execution of the present job segment would be jeopardized as a result of the previous job segment not being completed. At the completion of a post-mortem dump, the System Core-Storage Dump Program sets bit 17 to a 1 and stores the word "IBSXEC" in the SYSGET location before returning control to the System Supervisor. The Nucleus routine SYSRPT clears bit 17 to 0 at the beginning of each job.

The IBJOB monitor sets bit 16 of SYSJOB to a 1 to indicate that the Debugging Postprocessor is to be called. The IBJOB monitor regains control after an execution involving Debug by presetting the contents of SYSGET to "IBJOBb."

The address portion of SYSJOB contains a count of the number of jobs processed on the current input file. This count is maintained by the System Supervisor and is used to locate a specified job when the $RESTART

± control card is processed. The subsystems are in no way concerned with the job count.

Recognition of System Supervisor Control Cards by Subsystems

Each subsystem must recognize and act upon the $IBSYS, $EXECUTE, $STOP, $ID, and $JOB control cards. The action that must be taken by a subsystem when each card is recognized is as follows:

$IBSYS Card

When a $IBSYS card is recognized by a subsystem, it must return control to the System Supervisor. Whenever a subsystem gains control from the System Supervisor, the SYSGET location will contain the word "IBSYST." Therefore, it is not necessary for the subsystem to load the word "IBSYST" into the SYSGET location when a $IBSYS card is recognized.

$EXECUTE Card

When a $EXECUTE card is recognized by a subsystem and the subsystem name on the $EXECUTE card is the name of the subsystem, the subsystem retains control and continues normal processing. Otherwise, the subsystem stores the name specified on the card into the SYSGET location and returns control to the System Supervisor by way of SYSRET. If the subsystem had changed an availability chain, it would have previously set the sign of SYSJOB to minus.

$STOP Card

When a subsystem recognizes a $STOP card, whether or not it is located in the proper sequence on the input file, it loads the word "STOPb" in the SYSGET location and returns control to the System Supervisor by way of SYSRET.

$ID Card

When a subsystem recognizes a $ID card, it must TSX to SYSIDR as follows:

```
TSX      SYSIDR,4
PZE      L($ID)
     return
```

where L($ID) is the location of the first word of the buffer containing the $ID card in BCD form.

$JOB Card

When a subsystem recognizes a $JOB card, whether or not it is located in the proper sequence on the input file, it loads the word "IBSBSR" into SYSGET and then executes a TSX SYSRPT, 4 instruction. The SYSRPT routine in the System Nucleus will then determine if the System Supervisor must restore unit assignments (the sign of SYSJOB is minus) or control a between-jobs interrupt (sense switch 1 is DOWN). If the System Supervisor must do either, control is passed to the System Supervisor by way of SYSRET. Otherwise, control is returned to the subsystem. When the subsystem regains control, it must restore the word "IBSYST" in the SYSGET location and TSX to SYSIDR as follows:

```
TSX      SYSIDR
PZE      L($JOB)
     return
```

where L($JOB) is the location of the first word of the buffer containing the $JOB card in BCD form.

If a $JOB card is read from the card reader by a subsystem, the System Monitor receives control and prints a $JOB card on- and off-line in the following format:

```
1              16
$JOB           xxxxxx
```

The variable field is printed as xxxxxx instead of the actual text because (1) there is no common buffer between the subsystems and the System Monitor, and (2) the card cannot be reread, since backspacing is not possible on the card reader.

Input/Output Executor

The Input/Output Executor (IOEX) consists of a trap supervisor and a number of utility routines that are used in common by the System Monitor and the subsystems operating under its control. Any subsystem that is incorporated under the System Monitor and employs 729 Magnetic Tape Units, card equipment, direct access storage, or 7340 Hypertape Drives should use IOEX to ensure centralized control of input/output activity. The use of a single trap supervisor by the subsystems and the System Monitor not only minimizes input/output coding but also (1) ensures proper coordination of trapping, (2) enables a running log to be kept of tape positions, (3) enables error recovery procedures to be standardized, and (4) simplifies the diagnosis of input/output failures.

A subsystem communicates with IOEX and calls IOEX subroutines by way of a Communication Table located in storage just forward of IOEX. Location SYSIOX in the Communication Region of the System Nucleus (Appendix A) contains the address and length of this table. An entry in the IOEX Communication Table may be referred to by its symbolic address when using either the Macro Assembly Program (MAP) or the FORTRAN II Assembly Program (FAP). However, the symbols used to represent the addresses of the table entries for MAP are different from the symbols used to represent the same addresses for FAP. Both the MAP and FAP symbolic addresses, together with the function of each entry in the IOEX Communication Table, are listed in Figure 17 at the end of this section. When these entries are referred to symbolically using FAP, a Save Symbol Table (SST) pseudo-operation must be included in the first card group of the FAP source program. When these entries are referred to symbolically in a MAP absolute assembly, without definition in the source program, the MONSYM or JOBSYM option must appear on the SIBMAP control card. (A MAP relocatable assembly leaves these symbols as undefined virtual symbols, which IBLDR defines from symbols in the library.)

The following description of IOEX, together with a symbolic listing of the IOEX portion of the System Monitor, should provide the system programmer with the information required to use IOEX. In both the description and the listing, the MAP symbols are used when referring to entries in the IOEX Communication Table. The equivalent FAP symbols may be obtained by reference to Figure 17.

Unit Control Blocks

Each input/output device that may be referred to by the IBSYS Operating System is represented in IOEX by a four-word unit control block. The unit control blocks are generated by the System Monitor at initial start in accordance with assembly parameters. The following describes the format and contents of the unit control blocks for 729 Magnetic Tape Units, card equipment, direct access modules, and 7340 Hypertape Drives.

Unit Control Block for 729 Tape Units and Card Equipment

Each 729 Magnetic Tape Unit and each card unit is represented in IOEX by a four-word unit control block whose format is shown in Figure 9.

	S	1	2	3	4	5—17	18	19	20	21—35
UCW 1	A	M	T	R	C	Unit Address	EOT	DO	DI	Chain Address
UCW 2	S	R		SEL						
UCW 3	N			File Count			Record Count			
UCW 4										

Figure 9. Unit Control Block for 729 Tape Units and Card Equipment

The contents of each unit control block is interpreted as follows:

Word 1

A: Availability Flag
M: Attachment Flag

A=0: The unit is assigned to a particular function, not necessarily a system unit function, and it is not in the availability chain.

M=0: The assigned function of the unit is such that it should be repositioned when restarting.

M=1: The unit should not be repositioned when restarting (for example, if the unit is assigned as SYSOU1, SYSOU2, SYSPP1, SYSPP2, SYSCK1, or SYSCK2).

A=1: The unit is not assigned to any particular function.

M=0: The unit is attached to the channel and is in the availability chain. Card equipment is never in the availability chain.

M=1: The unit is detached from the channel and cannot be used.

If the user desires to assign a unit in the availability chain to a function not included in the System Unit Function Table, he should set the availability flag to zero, remove the unit from the availability chain, and set the attachment flag to indicate whether or not the unit should be repositioned when restarting.

T: Unit Type (for 729 tapes only)
- T=0: Model II or V
- T=1: Model IV or VI

R: Reserve Status Flag (intersystem use only)
- R=0: The unit is not reserved.
- R=1: The unit is reserved. Address bits 24-35 of word 1 contain data (two characters) for intersystem pickup. The unit should not be assigned a system unit function or be in the unit availability chain.

C: Channel Type
- C=0: 7607 channel
- C=1: 7909 channel

Unit Address: The address of the input/output unit is contained here. If the unit is a tape, the address is the BCD mode address; e.g., 1201 for 729 tape unit A1.

EOT: End-of-Tape Flag
- EOT=0: No end of tape has occurred on this unit.
- EOT=1: An end of tape has occurred while writing on this unit.

DO: Tape Density at Load Point
- DO=0: Low Density
- DO=1: High Density

DI: Density at the Current Tape Position
- DI=0: Low Density
- DI=1: High Density

Chain Address: This is the address of word 1 of the next unit control block in the availability chain. The chain address of the last unit is zero. This area is available to the user when the unit is not in the availability chain, or is not a reserve unit.

Word 2

S: Select Type
- S=0: Read
- S=1: Write

R: Permanent Redundancy Message (control)
- R=0: A message is printed if a permanent read redundancy occurs.
- R=1: No message is printed in the event of a permanent read redundancy.

SEL: Select Routine
SEL represents symbolically the location of a user's select routine, which initiates data transmission and the posting of completed input/output activity.

Word 3

N: Noise Record Flag (reading only)
- N=0: No noise records have been detected while reading.
- N=1: One or more noise records have been detected while reading.

File Count: The file count reflects the number of file marks written on or read from this tape.

Record Count: The record count reflects the number of records which have been written on or read from the current file.

Word 4

This word is provided for all-purpose systems usage. It is specifically used by IOCS labeling routines for storing the tape reel serial number in case of multifile reels.

Notes

The EOT flag and noise record flag are turned off only when the tape is returned to the rewound position.

The record count is complemented when backspacing from an end of file. A backspace which repositions in front of a file mark gives a record count (18-35) of 777777_8. For example, when writing occurs from such a position, the two low-order tag bits are cleared to prevent a spurious increase in the file count when the record count is increased.

The shaded area in Figure 9 is available to IOCS, or any subsystem using IOEX only.

Disk Unit Control Block

Each 1301/2302 Disk Storage Module is represented in IOEX by a four-word unit control block whose format is shown in Figure 10.

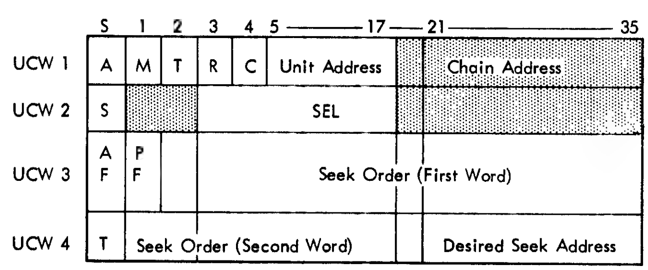


Figure 10. Disk Unit Control Block

The contents of each unit control block are interpreted as:

Word 1

This word has the same interpretation as for word 1 of the 729 tape unit control block except that when a disk module is attached to the channel, the System Monitor does not remove its unit control block from the availability chain even though it is assigned a system unit function. However, if input/output is performed on the unit by a subsystem which uses IOCS, the chain address of word 1 will be modified, thereby destroying the availability chain for the channel until it is re-established at the next $JOB or $RESTORE card. In general, the availability chain on a disk channel is of limited use, since unit assignment is based on cylinders rather than modules. Also, if the 7631 File Control Unit is a Model III or IV, bit position T will be one (T=1); otherwise, position T equals zero (T=0).

The disk unit address has the format shown in Figure 11.

5 ——— 8	9 ——— 11	12	13	14 ——— 17
Channel (1 through 8)	Device (1 for 1301 or 7 for 2302)	Data Channel Switch (0 or 1)	Access (0 or 1)	Module (0 through 9)

Figure 11. Format of Disk Unit Address

Examples:

Channel C, 1301 Disk, Access 0, Module 1, Data Channel Switch Setting 1: address is $(3101)_8$

Channel E, 2302 Disk, Access 1, Module 9, Data Channel Switch Setting 2: address is $(5771)_8$

Word 2

The interpretation of this word is the same as for word 2 of the 729 tape unit control block except that redundancy message control is not provided. Any disk error will produce an on-line message if recovery is unsuccessful.

Word 3

AF: Seek Request Flag

AF=0: No seek is requested, or an ATTENTION signal was received on the previously requested seek.

AF=1: A seek is requested by the user, or a seek was issued and an ATTENTION signal is awaited. This flag is set to 1 by the user when a seek is requested. It is reset to 0 by IOEX when the ATTENTION signal is received.

PF: Seek Issued Flag (used by IOEX only)

PF=0: No seek was issued.

PF=1: A seek was issued and an ATTENTION signal is awaited.

Seek Order: The six bytes of word 3 and the first two bytes of word 4 are used to form the seek order for any seek requested of IOEX by the user. The last two bytes of word 3 and the first two bytes of word 4 will contain the track address (in BCD) for the desired seek.

Word 4

T: Track Flag

T=0: IOEX will set up the BCD track address for a requested seek, obtaining its information from the binary track address specified by the user in the address of word 4.

T=1: The user, on requesting a seek, has already set up the BCD track address in words 3 and 4. T is reset to zero by IOEX after the seek is issued.

Desired Seek Address: This is the track address (binary) for a seek requested by the user. The address of word 4 is never destroyed by IOEX.

Drum Unit Control Block

Each 7320 Drum Storage unit is represented in IOEX by a four-word unit control block, which is identical in format and usage to a Disk UCB except for the unit address in bits 5 through 17 of word 1.

The drum unit address has the format shown in Figure 12.

5 ——— 8	9 ——— 11	12	13	14 ——— 17
Channel (1 through 8)	Device (3 for drum)	Data Channel Switch (0 or 1)	Access (0)	Module (0, 2, 4, 6, or 8)

Figure 12. Format of Drum Unit Address

Examples:

Channel C, Access, Module 0, Data Channel Switch Setting 0: address is $(3300)_8$

Channel F, Access, Module 8, Data Channel Switch Setting 1: address is $(5350)_8$

Hypertape Unit Control Block

Each Hypertape drive is represented in IOEX by a four-word unit control block with the format shown in Figure 13.

	S	1	2	3	4	5 ——— 17	18	19	20	21 ——— 35
UCW 1	A	M		R	C	Unit Address	EWA			Chain Address
UCW 2	S	B		SEL						
UCW 3	AF	PF	I	File Count			Record Count			
UCW 4										

Figure 13. Hypertape Unit Control Block

The Hypertape unit control block is similar to the unit control block for 729 tapes. The differences between a Hypertape unit control block and a 729 tape unit control block are as follows:

Word 1

Unit Address: The unit address field for Hypertape is similar to that for disk storage. It has the format shown in Figure 14.

5—8	9—11	12	13	14—17
Channel (1 through 8)	Device (0 for Hypertape)	Data Channel Switch (0 or 1)	0	Unit Number. (0 through 9)

Figure 14. Format of Hypertape Unit Address

Examples:

Channel C, Hypertape Drive 1, Data Channel Switch setting $1=(3001)_8$

Channel H, Hypertape Drive 9, Data Channel Switch setting $2=(10051)_8$

EWA: End Warning Area Flag

EWA=0: The unit is not in the EWA.

EWA=1: The unit is or has been in the EWA.

The EWA flag is turned off only when the tape is returned to the BOT marker; or whenever HCCR, HCHC, or HUNL is executed.

Word 2

R: Select Mode Flag

R=0: Normal select

R=1: Select for backward reading

Both this bit and the S bit are set by the user.

Word 3

AF: AF Flag

PF: PF Flag

These flags are used only by IOEX to control the use of Attention servicing routines.

General Use of IOEX and Unit Control Blocks

The following steps describe the general procedure to be taken by the user when input/output activity for a unit is desired.

Before activity can be requested on an input/output unit, the user must determine whether or not the unit is free for use. This is accomplished by waiting for word 2 of the unit control block to become zero. Once it becomes zero, the user must assume control of the unit by placing the location of his select routine in the decrement of word 2. The user is also responsible for setting bit positions S and R in unit control block word 2. R is omitted for direct access storage units.

After the input/output unit is secured for use, the user may request channel activity for the unit by entering the IOEX routine .ACTV. This routine, which is used to handle data select activity, permits all necessary entries into the user's specified select routine as soon as the channel is free to accept input/output activity from the unit. Non-data select activity is handled by the routine .NDSEL.

On the first entry to the select routine, called the (SEL+) or select entry, the user must initiate execution of the input/output commands for the unit. Termination of the commands must always set up a trap condition for the CPU.

On the last entry to the select routine, called the (SEL−) or posting entry, input/output activity for the unit is complete. At this time the user may relinquish control of the unit by setting word 2 of the unit control block to zero, or he may choose a number of other options using information described under "Posting Entry."

The user distinguishes between (SEL+) and (SEL−) entries to his select routine by testing the sign of the accumulator at time of entry.

Normally, any direct reference to a unit control block at other than trap (SEL) time should be trap-protected by the sequence:

```
        ENB     L(0)
         .
         .
         .
        ENB*    .TRAPX
```

Data Transmission Via Select Routine

The initiation of data transmission operations for a unit, and the maintenance of any request queue for a unit, is the responsibility of the user of IOEX. These functions must be provided in a subroutine, labeled SEL for example, which is entered twice for each data transmission operation that results in a trap. The calling sequence from IOEX is:

	TSX	SEL,4	
Return 1			Normal return at either entry
Return 2			Returns used at (SEL−)
Return 3			time because of errors indicated. (See appropriate recovery section.)

Upon either entry to the select routine, IOEX disables channel trapping and provides:

C(IR1)	The 2's complement of channel index (0=A, 1=B, etc.)
S(AC)	Sign of accumulator: plus for (SEL+) and minus for (SEL−)
A(AC)	Location of the unit control block

Select Entry

If the sign of the AC is plus on entry to SEL, the routine must initiate a data select command sequence terminating in a trap. Traps must not be enabled by (SEL+). This is taken care of by IOEX. For 7909 channel operation, the command sequence should terminate with a

```
        TCH     SYSTWT
```

which will produce the necessary trap. In addition, an SMS command which disables ATTENTION interrupts and enables UNUSUAL END interrupts must initiate the command sequence.

For Hypertape, on each entry to the (SEL+) routine, the user must initiate a data select command which will

read or write one record. This restriction, which is similar to the restriction on the use of 729 tape units, is necessary to ensure the correct record count in word 3 of the unit control block.

Posting Entry

If the sign of the accumulator is negative on entry to SEL, a trap has occurred as a result of the previous select for the unit. The following information is furnished upon entry:

Sense Indicators—7607 Channel

Bit S Noise record flag
- 1 The record was *not* an apparent noise record.
- 0 The record was an apparent noise record.

Bit 1 End of file (read) or end of tape (write)
- 0 No end of file or end of tape.
- 1 End of file or end of tape.

Bit 2 Permanent redundancy (read only)
- 0 No permanent redundancy.
- 1 Permanent redundancy.

Bit 3 Read or write indication
- 0 Read.
- 1 Write.

Bit S is on in all cases except when a noise record is detected.

Sense Indicators—7909 Channel, Disk and Drum

Bits S, 1-5	7909 Control counter
Bit 6	Input/output check
Bit 7	Sequence check
Bit 8	UNUSUAL END
Bit 9	ATTENTION on Interface 0
Bit 10	ATTENTION on Interface 1
Bit 11	Interface check
Bits 12-35	First four bytes of the 7631 sense data. Bits 12-35 are supplied only if the trap was a result of an UNUSUAL END. In such a case, bit 8 will be a 1.

Sense Indicators—7909 Channel, Hypertape

Bits S, 1-5	7909 Control Counter
Bit 6	Input/output check
Bit 7	Sequence check
Bit 8	UNUSUAL END
Bit 9	ATTENTION on Interface 0
Bit 10	ATTENTION on Interface 1
Bit 11	Interface check
Bits 12-35	Bytes 1, 3, 4, 5, 6, and 7 (packed) of the 7640 sense data: supplied only if the trap was the result of an UNUSUAL END signal, i.e., if bit 8 is a 1.

Cell .COMM

This word contains the results of a store channel instruction for the 7607/7909 channel. If an UNUSUAL END condition occurs in the 7909 channel, the contents of .COMM will be overlaid with the contents of the address counter (3-17) and the command counter (21-35).

URRX,1 Table

The URRX,1 Table is used to keep a count of the number of redundancies on the channel.

Redundancy Counts:

PZE N1,,N2

N1:	Number of recovery entries to (SEL+).
N2=0:	No permanent redundancy (while reading) or no erase areas (writing).
N2=1:	Permanent redundancy (reading) or one or more erase areas (writing). If a permanent redundancy, N1=0. N2 is always zero for disk, drum, and Hypertape input/output.

The URRX,1 Table may be referenced by indirectly addressing the communication cell .URRX. IR1 must contain the complement of the channel index.

If EOT (or EWA if Hypertape) is detected while writing, the EOT bit is set ON in the unit control block, and the EOT indication is given on each subsequent entry to (SEL−) for the unit until the tape is rewound. When an end-of-file indicator is given to (SEL−), the tape position has already been adjusted. The communication location .LTPOS contains the tape position prior to this adjustment. A redundancy indication cannot occur together with an end-of-file indication.

Design of Select Routines

Select routines must not destroy the contents of IR1. IR2 and the sense indicators need not be saved. They must not modify or change the unit address in word 1 of the unit control block or the four high-order bytes (except the sign bit) of word 3 of a direct access storage unit control block.

In disk and drum usage, the select routine must interpret word 1 of the unit control block if use of the compact access and module bits are needed for setting up Prepare to Verify orders, or it must use .FDAMT. (See "Form Disk/Drum Order" under "IOEX Utility Routines.")

Select routines should be designed to minimize processing time while the machine is trapped.

Unit Priority on a Channel

Channel Priority Location

When activity begins on a unit of a given channel, IOEX places the address of the unit control block in a channel priority location. This allows the unit to retain priority until all its waiting operations have been completed. IOEX clears this location when the user clears word 2 of the unit control block.

Upon a normal return from a (SEL−) routine, IOEX selects the next unit to be activated by examining the channel priority cell. The same unit will be reselected if control was not relinquished by the user by clearing word 2 of the unit control block.

Use of Channel Priority Location

If the channel priority location is not zero, its address portion is interpreted as the location of the unit control block for the unit to be activated next. Word 2 of the unit control block is then tested.

If word 2 is not zero, (SEL+) for the unit is entered. If it is a direct access storage unit control block, the (SEL+) entry is held up (return is still made to the user) until any pending seek for the unit has been issued and attention received.

If word 2 of the unit control block is zero, the priority cell is cleared and the channel is scanned for a waiting unit, a unit that has a SEL routine specified in word 2 of its unit control block. In addition, if it is a unit control block for direct access storage, the AF flag in word 3 must be zero. If a waiting unit is found, the location of its unit control block is placed in the channel priority cell, and (SEL+) is entered for the unit. If none is found, the channel is allowed to become dormant. In the case of a direct access storage unit, however, any pending seeks are issued, and the channel becomes dormant only if no seeks are waiting to be issued.

Activating a Channel and/or Assigning Priority

A channel which is dormant is activated, or a unit is given top priority on a channel, by means of the routine .ACTV. The calling sequence is:

```
TSX     .ACTV,4
P       A,T
    return
```

where the parameter A, T gives the address of a location (possibly a System Unit Function Table entry) which contains:

```
OP      UCB,X,Y
```

UCB is the address of the unit control block of the unit desiring input/output activity. OP, X, and Y are ignored by .ACTV. Note the double indirectness to the actual unit address.

When P is PZE in the calling sequence, controls are set up so that the user's select routine is entered whenever the channel is free to accept activity on the specified unit. From the discussion under "Unit Priority on a Channel," it is obvious that no action by .ACTV is necessary if the channel is active upon entry. If such is the case, control is immediately returned to the user.

If the channel is dormant, .ACTV enters the user's (SEL+) routine directly before returning. In direct access storage, the AF flag in word 3 of the unit control block must be zero for (SEL+) to be entered. If the AF flag is not zero, .ACTV first activates the channel by issuing a seek for the access and for any other accesses on the channel requesting the issuance of a seek. The select routines are entered as the ATTENTION signals are received.

This entry to .ACTV, requesting the issuance of seeks (P=PZE), is the only entry to the routine which does not require a SEL routine to be specified in word 2 of the unit control block. In such a case, .ACTV will still issue all requested seeks. This allows the user, if he so desires, to use "seek time" with the knowledge that the seek ATTENTION signal will be recorded when received by IOEX but that no select routine for the unit will be entered. When the user is ready to have the select routine entered, he places its location in word 2 of the unit control block and, to ensure that the channel is activated if it became dormant, reenters .ACTV with the AF flag equal to zero. To determine if a direct access storage unit is free for use when this option is exercised, the user must test the AF flag as well as word 2 of the unit control block for zero.

When P is MZE in the calling sequence, the indicated unit is given top priority on the channel; that is, the channel priority cell is set for the unit, and (SEL+) is entered as soon as possible. If the entry to .ACTV is made during non-trap time, control is not returned to the user until the (SEL+) entry for the unit has occurred. In direct access storage usage, if the AF flag in word 3 of the unit control block equals one (AF=1), the necessary seek is issued prior to the (SEL+) entry.

If the entry to .ACTV is at trap time, e.g., during (SEL−), the unit is given top priority by the setting of the channel priority cell. However, return is made immediately. Hence, a subsequent entry to .ACTV requesting priority (P=MZE), cancels the priority effects of the previous entry if it occurs before (SEL+) is entered for the previous entry.

Any entry to .ACTV at trap time must be made with MZE in the calling sequence, i.e., P=MZE. Also, the entry must be for a unit connected to the channel that caused the trap.

The .ACTV routine always enables traps on all channels upon return, unless entry is from a select routine.

.ACTV may not be entered during (SEL+) time.

.ACTV makes a validity test on the specified input/output unit. To be valid, the specified location of the unit control block must fall within the range of core storage provided for all unit control blocks. In addition, the A flag of word 1 of the unit control block must be zero. If a unit is judged to be invalid by these criteria, an automatic post-mortem dump is taken after the following message is printed on the System Printer:

ILL UNIT REQ'ST AT xxxxx

After the dump is taken, the System Supervisor skips to the next job segment.

In FAP, the above validity test may be bypassed by using:

```
        TSX        (ACTVX,4
```

in lieu of:

```
        TSX        (ACTIV,4
```

in the calling sequence, (ACTIV being the FAP equivalent of .ACTV.

The routine .ACTV has the same purpose and use for Hypertape as it has for 729 tape units. Namely, upon entry to .ACTV, a SEL routine must be specified in the appropriate unit control block. Entry is essentially a request by the user for input/output activity for the unit indicated in the calling sequence.

Non-Data Selects

Non-data selects are executed by the routine .NDSEL. The calling sequence is:

```
        TSX        .NDSEL
        PZE        A,T,NDS
            return
```

The parameter A,T has the same meaning as in the calling sequence for .ACTV. NDS is interpreted for 729 tape units as follows:

NDS=0	NOP
NDS=1	SDNL
NDS=2	SDNH
NDS=3	REW
NDS=4	RUN
NDS=5	BSR
NDS=6	BSF
NDS=7	WEF

.NDSEL gives the specified unit top priority by using .ACTV with MZE in the calling sequence, i.e., P=MZE. Hence, the non-data select is executed as soon as the unit's present activity is complete. Return is made only after the non-data select has been executed .NDSEL may not be entered during (SEL+) time.

The .NDSEL routine always enables traps on all channels upon return unless entry is from a select routine. .NDSEL may be entered at trap time only for a unit which is on the channel that has trapped. Non-data selects for card, disk, and drum equipment are ignored.

.NDSEL does a validity test on the specified unit control block in the same manner as .ACTV. In FAP, the test may be bypassed by using TSX (NDSLX rather than TSX (NDATA. Note that a NOP (NDS=0) entry to .NDSEL will have the effect of performing a validity test and nothing else. Backspacing a record across the previous file mark complements the record count in bits 18-35.

WEF causes the following sequence to be executed:

```
        WEFX
        TCOX*
        TRCX
        ETTX
```

Recovery is attempted if a redundancy occurs while writing an end-of-file mark. The tape is backspaced, and the file mark is rewritten and checked as often as necessary. If the EOT condition is detected on 729 tape after a nonredundant end of file, return is made to 2, 4. *The normal return for WEF is 3, 4.*

Hypertape orders are also handled by the .NDSEL routine. The calling sequence is:

```
        TSX        .NDSEL,4
        PZE        A,T,NDS
            return
```

The parameter A, T is the same as for 729 tape units. NDS is interpreted as follows:

NDS=0	NOP((NDATA NOP)
NDS=28	HCCR—Change Cartridge and Rewind
NDS=30	HRWD—Rewind
NDS=31	HRUN—Rewind and Unload
NDS=32*	HERG—Erase Long Gap
NDS=33*	HWTM—Write Tape Mark
NDS=34*	HBSR—Backspace Record
NDS=35*	HBSF—Backspace File
NDS=36*	HSKR—Skip Record
NDS=37*	HSKF—Skip File
NDS=38	HCHC—Change Cartridge
NDS=39	HUNL—Unload
NDS=42	HFPN—File Protect On

Those orders marked with an asterisk are handled in the same way as non-data selects for 729 tape units. That is, the .NDSEL routine uses .ACTV+1 with an MZE in the calling sequence. Hence, the order is executed as soon as the unit's present activity is complete. Return is made from the .NDSEL routine only after the order has been executed.

The remaining orders will cause an ATTENTION signal upon completion of the operation. They are treated in the following manner: After any present activity is finished, IOEX executes the order by specifying a select routine. Upon receiving the applicable ATTENTION signal, IOEX clears word 2 of the unit control block. The user must not ask for new activity on that unit until word 2 is cleared.

All orders must be given through the .NDSEL routine. The return from the .NDSEL routine is always to 2, 4. Entry to the .NDSEL routine during trap time is permitted only for those Hypertape orders marked with an asterisk, and for those units on the same channel and Data Channel Switch setting as the unit which trapped.

Redundancy Recovery

Writing on 729 Tapes

The redundancy trapping mode is used for a write operation. If the first attempt to write a record produces a redundancy trap, the following procedure is followed:

1. The tape is backspaced one record.
2. An erase area is written. If this operation produces a redundancy check, an operator message is printed.

3. The record is rewritten (that is, (SEL+) is entered) and checked for redundancy.

4. Steps 1-3 are repeated until the record is written correctly or the EOT condition is sensed. After each group of 25 erase operations, an operator message is printed.

IOEX stops when the EOT condition occurs during redundancy recovery. However, there is always at least one erase-rewrite sequence attempted, and, if the rewrite is successful, the stop does not occur.

If, following the successful writing of a record, IOEX determines that the apparent record length was less than three words, an entry is made to (SEL−) with the noise record condition indicated in the sense indicators. Two exits are available for this condition:

Return 1	The record is accepted.
Return 2	The record is accepted, and an operator message is printed indicating that a short record has been written.

The record count is increased before entry to (SEL−). It is not increased before rewrite entries to (SEL+).

On each rewrite entry to (SEL+) during redundancy recovery, URRX,1 has the following configuration:

```
PZE     N,,1
```

where N is the number of consecutive erase areas which have been written. Following a successful redundancy recovery, on entry to (SEL−), URRX,1 has the same configuration. N in this case is the total number of erase areas written on this recovery.

Reading from 729 Tapes

The redundancy trapping mode is not used during read operations so that a full word count of record size may be secured. If the redundancy occurs as the result of a reading operation, the following steps are taken:

1. If the record is an apparent noise record, (SEL−) is entered with appropriate indication in the sense indicators. If the record is not an apparent noise record, step 2 is taken.

2. The redundancy count (address of URRX,1) is increased by 1. It is initially zero.

3. If all of the following three conditions exist, a tape cleaner action is taken. If one or more conditions do not exist, step 4 is taken.
   a. The redundancy count is 1 or a multiple of 10 plus 1.
   b. The noise record bit in word 3 of the unit control block is off.
   c. There are at least two previous records in the current file.

The tape cleaner action consists of backspacing over the redundant record and the two previous records, giving two dummy reads, and then entering (SEL+) to reread the redundant record.

4. The tape is backspaced over the redundant record, and (SEL+) is entered to reread the redundant record.

5. Steps 1 through 4 are repeated until the record is read correctly or until the redundancy count reaches the value of the assembly parameter RDUNRT. If the value of RDUNRT is reached, a permanent read redundancy is assumed. If such is the case, the record count is increased by one and an operator message is printed if not suppressed by the control bit in word 2 of the unit control block. In addition, the instruction:

```
PZE     0,,1
```

is stored in URRX,1, and (SEL−) is entered with the permanent redundancy indication in the sense indicators.

Read redundancy checking and recovery may be suppressed by setting RCTX,1 to zero in (SEL +). The RCTX,1 Table may be referenced by indirectly addressing the communication location .RCTX. IR1 must contain the complement of the channel index. On every entry to (SEL+), RCTX,1 is set on (to nonzero).

Exits from (SEL—) for 729 Tapes

Three exits are available for (SEL−).

Return 1	The record is accepted.
Return 2	The record is considered noise and is discarded. An operator message is printed, and the noise record bit is set on in word 3 of the unit control block. The record count is reduced by 1; URRX,1 is cleared, and (SEL+) is re-entered to read the next record.
Return 3	The redundancy recovery procedure is entered at step 2.

After each successfully completed write operation, and after a redundant read operation, the following test is made for an apparent short (noise) record. The address of the last input/output command is subtracted from the address of the Store Channel word to obtain an apparent word count. If this word count is less than three, IOEX gives the noise record indication to (SEL−).

Obviously, there are sequences of input/output commands which will produce this noise record condition, even when the true record length is greater than two words.

No test is made for use of the indirect addressing feature of input/output commands. If this feature is used, a short record will not be detected.

Reading and Writing Disk and Drum

Figure 15 is a table showing the bit assignments for the first four bytes of the 7631 sense data. When an UNUSUAL END signal occurs for a condition indicated in bytes 3, 4, or 5 of the sense data, IOEX takes the following recovery action.

Recovery Action

If the UNUSUAL END occurred during a read or write on a disk, (SEL+) is re-entered up to four times.

Byte	Bit	Assignment	Abbreviation	Comment
1	2	Reserved	--	Summary Bits
	4	Program Check	PRGCK	
	5	Data Check	DATCK	
	6	Exceptional Condition	EXCND	
2	2	Invalid Sequence	I'SEQ	Program Check
	4	Invalid Code	I'CDE	
	5	Format Check	FRMCK	
	6	No Record Found	NOREC	
3	2	Invalid Address	I'ADR	Data Check
	4	Response Check	RSPCK	
	5	Data Compare Check	DCOMP	
	6	Parity or Cyclic Check	P/CCK	
4	2	Access Inoperative	INOPR	Exceptional Condition
	4	Access Not Ready	NTRDY	
	5	File Frame Circuit Check	DKCRK	
	6	File Adapter Circuit Check	CUCRK	

Figure 15. Bit Assignments for the First Four Bytes of the 7631 Sense Data

If the UNUSUAL END persists, two seeks are given to recalibrate the arm. This is followed by the original seek requested by the user. (SEL+) is then entered again. If the UNUSUAL END still persists, (SEL+) will be re-entered up to three more times. If success is not achieved, (SEL−) is entered with the proper indication.

If the UNUSUAL END occurred during a read or write operation on a drum, (SEL+) is re-entered up to four times. If the UNUSUAL END persists, (SEL−) is entered with the appropriate bits set in the sense indicators to indicate the error condition.

Select Exits

Whenever (SEL−) is entered because of an unsuccessful recovery attempt, an operator message will be printed indicating the error. The message will have the following form:

```
UNIT {xDam/s} WRITE ERROR–TRCK OOtttt
     {xNam/s} (READ)
x1,x2,x3, . . . ,x24
```

where tttt is the last track "seeked" by IOEX and x1, x2,x3,...,x24 are octal numbers representing all 72 bits of the two 7631 sense data words (Appendix C).

In addition, the address portion of URRX,1 will contain the total number of re-entries to (SEL+) made in attempting to recover. On each entry to (SEL+), URRX,1 will be zero or will contain the number of times (SEL+) has been entered previously in attempting to recover from an UNUSUAL END.

The three possible exits from (SEL−) are:

Return 1	Normal return.
Return 2	Not to be used.
Return 3	Re-enter (SEL+) and repeat the recovery actions if necessary.

On any entry to (SEL+), 7631 checking may be qualified by an exclusion flag corresponding to the 7631 sense data listed in Figure 15.

Procedure to Qualify 7631 Checking

Into cell RCTX,1 place a word of the following format:

```
        PZE     FLAGWD
```

where FLAGWD is the location of a word which is an exclusion mask to be tested by IOEX prior to any error procedures.

For example, set exclusion to ignore NOREC, RSPCK, DCOMP, and P/CCK.

```
            CLA     FLBITS
            STO*    .RCTX
             .
             .
             .
FLBITS      PZE     *+1
            OCT     060107000000
```

The proper summary bit is set on (to nonzero) as long as it is desired to exclude at least one of the errors to which it refers. In the above example, exclusion of the NOREC error also necessitated the exclusion of its summary (PRGCK) bit. On every entry to (SEL+), RCTX,1 will be initialized to zero.

Reading and Writing Hypertape

Figure 16 is a table showing bytes 1, 2, 3, 4, 5, 6, and 7 of the 7640 sense data which have been packed into four bytes as shown in the left-hand columns. The table indicates the associated recovery action taken by IOEX when an UNUSUAL END signal occurs other than on an order issued by the .NDSEL routine.

Action 1

The (SEL+) routine is re-entered. If the error persists, the (SEL−) routine will be entered and the proper sense data will be in the sense indicators.

Action 2

The (SEL−) routine is not entered with the error indication until the sequence HBSR (or HSKR) — Enter (SEL+) is repeated ten times without success.

Action 3

An HBSR order followed by an HERG order is executed. The (SEL+) routine is then re-entered. If the same UNUSUAL END signal persists after ten re-entries to (SEL+), the (SEL−) routine is entered with the proper indication.

Action 4

An operator message is printed indicating the corrective action to be taken by the operator. Upon completion of the necessary operator action, the (SEL+) routine is re-entered.

Byte	Bit	Assignment	Abbreviation	Recovery Action RD	Recovery Action WR	Comment
1	1	Operator Required	OPREQ			
	2	Program Check	PRGCK			Summary Bits
	3	Data Check	DATCK			
	4	Exceptional Condition	EXCND			
	5	Selected Drive Not Ready	NTRDY	4	4	Operator Required
	6	Selected Drive Not Loaded	NTLOD	4	4	
2	1	Selected Drive File Protected	FILPR	-	4	Program Check
	2	Not Used	-			
	3	Invalid Order Operation Code	I'ORD	1	1	
	4	Selected Drive Busy	DRBSY	1	1	
	5	Selected Drive at BOT	ATBOT	1	1	
	6	Selected Drive at EOT	ATEOT	1	1	
3	1	Correction Occurred	-			
	2	Channel Parity Check	PARCK	2	3	
	3	Code Check	CDECK	2	3	Data Check
	4	Envelope Check	ENVCK	2	3	
	5	Overrun Check	OVRCK	2	3	
	6	Excessive Skew Check	ESKCK	2	-	
4	1	Track Start Check	TRSCK	2	3	
	2	Not Used	-			
	3	Selected DR Read a Tape Mark	-	*		
	4	Selected DR in EWA	-		**	
	5	No Data Transmitted	NDTRN	2	1	Exceptional Condition
	6	Not Used	-			

* When a tape mark is read, return will be made to the (SEL-) routine with the proper sense data in the sense indicators. The applicable unit control block will contain a zero record count, and the file count will be increased by one. In addition, location .LTPOS will contain word 3 of the unit control block as it was before the tape mark trap occurred.

** When writing continues into the EWA, the (SEL-) routine will be entered with the proper sense data in the sense indicators. In addition, the appropriate bit in word 1 of the unit control block will be set on.

Figure 16. Recovery Action Taken on an Unusual End Signal

Exits

The three possible exits from the (SEL−) routine are:

Return 1	Normal return.
Return 2	Not to be used.
Return 3	Re-enter (SEL+) and repeat the recovery actions if necessary.

Whenever the (SEL−) routine is entered with an UNUSUAL END signal, an operator message is printed indicating the error. The message will have the following form:

```
UNIT xHk/s WRITE ERROR
           (READ)
x1,x2,x3, . . . ,x24
```

where x1,x2,x3,...,x24 are octal numbers representing all 72 bits of the two 7640 sense data words (Appendix D).

As with direct access storage, 7640 checking may be qualified upon entry to the (SEL+) routine by means of an exclusion flag corresponding to the packed 7640 sense data shown in Figure 16. Thus, the user can specify with flag bits any unusual conditions which he may wish to ignore. For example:

```
          CLA     FLBITS
          STO*    .RCTX
          .
          .
          .
FLBITS    PZE     *+1
          OCT     301720000000
```

will cause IOEX to bypass any recovery attempts if any one of the following conditions causes an UNUSUAL END signal:

I'ORD
DRBSY
ATBOT
ATEOT
PARCK

These exclusion bits will also prevent the printing of any on-line message which usually would follow one of the above errors.

Channel Control Tables

The (SEL+) routine can make use of the IOEX Tables of Reset and Load Channel instructions and Start Channel (7909) instructions. This should be done indirectly through the communication locations .RCHX and .STCXI. IR1 must contain the complement of the channel index.

The form of the tables themselves is:

```
RCHX  RCHA   **    STCX  STCA
      RCHB   **          STCB
.RCHX contains PZE  RCHX,1.
.STCXI contains PZE STCX,1.
```

If .RCHX is used, the applicable address in the RCHX Table must be set upon every entry to (SEL+).

IOEX Utility Routines

The following are IOEX utility routines that may be used by subsystems operating under control of the System Monitor. The routines .MWR, .PUNCH, .CVPRT, .PAUSE, .PAWS, .BCD5R, and .BCD5X all enable trapping (except at trap time) upon exit. This trapping can be suppressed by the user by setting the address portion of the IOEX Communication Table entry .ENBSW to nonzero. The address must be reset to zero by the user.

Message Writer

Messages can be printed both on-line and off-line, using the subroutine .MWR. Off-line messages are actually printed by a separate subroutine, SPOUT, which is called by .MWR when an off-line message is specified in the .MWR calling sequence. SPOUT is stored in core storage locations SYSEND-199 through SYSEND. Therefore, if any subsystem uses SPOUT, it must consider SYSEND-200 as the end of usable core storage, rather than SYSEND.

The calling sequence for .MWR is:

```
TSX     .MWR,4
PFX     N
P       L1,T1,M1+512*SPR1
P       L2,T2,M2+512*SPR2
.
.
.
P       LN,TN,MN+512*SPRN
```

If PFX=PZE, the message is printed on-line and no reference to SPOUT is made. A subsystem that destroys SPOUT must specify PZE whenever .MWR is used. PFX is interpreted as PZE for any call to .MWR during (SEL+) or (SEL−) time.

If PFX=MZE, the message is recorded both off-line and on-line.

If PFX=MON, the message is recorded off-line only.

N is the number of entries following the calling sequence. M words (six characters each) beginning in location L, T are converted and placed in the line image for printing. If P=PZE, the image is taken to be complete and the line is printed. If P=MZE, this line is considered incomplete, and the L,T,M of the next calling sequence entry are used to continue building the image, beginning with the next print position to the right. If the number of words specified for a particular line is greater than 12, only the first 12 are printed on-line.

The sense exit SPR, if given, is activated either before or after the line is printed, depending on whether P equals MZE or PZE. (SPOUT will ignore an SPR appearing in a word with P=MZE.) For SPOUT, SPR must be 0 (single space), 1 (eject), or 4 (double space). To activate an exit before printing the first line, an entry of the form

```
PZE     **,,512*SPR
```

may be used. This prints a blank line, followed by an activation of the hub SPR. The same entry with the prefix changed to MZE may be used preceding another calling sequence word to activate the hub SPR (on-line only) without printing a blank line.

.MWR can be used either at trap time or at non-trap time. Printing is immediate; that is, the printing operation is started before the return from .MWR. Printing at trap time will be on-line only.

Alphameric Punch

The entry

```
TSX     .PUNCH,4
```

with sequence similar to .MWR, excepting sense exits, provides for punching BCD cards on-line for accounting, etc.

Error Pause

The instruction

```
TSX     .PAWS,4
```

causes a machine stop (HPR −1) after on-line printing of the message

PRES STRT TO GO ON

Pressing the Start key causes on-line printing of the message

... CONTINUING

and returns control to 1, 4.

Operator Action Pause

The instruction

```
TSX     .PAUSE,4
```

causes a machine stop (HPR −1) after on-line printing of the message

OPER. ACTION PAUSE...

Pressing the Start key causes on-line printing of the message

. . . CONTINUING

and returns control to 1, 4.

BCD Zero Conversion

The instruction

```
XEC     .IBCDZ
```

replaces decimal zeros in the BCD number located in the MQ with BCD zeros, i.e., 12_8. A CRQ is peformed. This routine is used for making disk/drum orders only.

Form Disk/Drum Order

The following routine places AMTTTT disk or drum data into the location of a specified order. It also sets interface bits for SMS and places RR (HA2) into the second word of the order.

The calling sequence is:

```
TSX     .FDAMT,2
BCI     1,00RR0B
PZE     DORDER,T
    return
```

IR4 equals −L (UCB), and bytes 3 through 6 of the MQ contain TTTT head and track in BCD. DORDER,T is the location of the disk/drum order. RR is HA2 identifier, and B is the mask for the 7909 SMS command. Bit 35, i.e., the low-order bit of B, is set for interface selection by .FDAMT. T may be zero or IR1.

Post-Mortem Dump

The instruction

```
TSX     .STOP,4
```

causes a transfer to the location SYSDMP in the System Nucleus. A transfer to SYSDMP causes a post-mortem dump of core storage followed by the skipping of cards on the system input file until a $IBSYS, $EXECUTE, $JOB, or $STOP card is encountered.

Symbolic Unit Conversion

The instruction

```
TSX     .SYMUN,4
```

converts the unit address located in the decrement of the MQ, i.e., LDQ with the address of word 1 of the unit control block, to its BCD equivalent as it appears on a $ATTACH card. The results, straddled by any necessary BCD blanks, are located in positions P, 1-35 of the AC upon return to 1,4.

Binary to Decimal Conversion — AC Decrement

The instruction

```
TSX     .DECVD,4
```

converts the binary number in the decrement of AC to its BCD equivalent. The results are located in the low-order end of the MQ. The high-order MQ character is a BCD blank. Control is returned to 1,4.

Binary to Decimal Conversion — AC Address

The instruction

```
TSX     .DECVA,4
```

has the same function as .DECVD, except that the address portion of AC is converted.

Binary to BCD Octal Conversion — MQ Decrement

The instruction

```
TSX     .BCD5R,4
```

converts the binary number in the decrement of MQ to its octal equivalent in BCD code. The results are located in the low-order end of the AC. The high-order AC character is a BCD blank. Control is returned to 1,4.

Binary to BCD Octal Conversion — S, 1-14 of MQ

The instruction

```
TSX     .BCD5X,4
```

has the same functions as .BCD5R above, except that bit positions S, 1-14 of the MQ are converted.

Convert and Append Unit Designation to Message

The words "UNITXXXXXX," where XXXXXX is converted from the unit address in D(MQ), i.e., LDQ with the address of word 1 of the unit control block, can be appended to a message by the calling sequence

```
TSX     .CVPRT,4
PFX     L,T,M+512*SPR
    return
```

where PFX is interpreted the same way as the PFX directly following the TSX .MWR,4. The remainder of the control word, that is, L,T,M+512*SPR, is interpreted the same as in the control words in .MWR.

Freeing a Channel

The user tests channel activity with the sequence

```
ZET*    .CHXAC
TRA     *-1
```

IR1 must contain the 2's complement of the channel index (0 = channel A, 1 = channel B, etc.).

When the ZET falls through, all activity on the channel is complete, including any disk-seek, drum-seek, or Hypertape free-running orders.

MAP Symbolic Address	FAP Symbolic Address	IOEX Entry		Function
.ACTV	(ACTIV	TTR	TEST	Activate Routine and Test
	(ACTVX	TTR	ACTIV	Activate Routine Without Test
.NDSEL	(NDATA	TTR	TEST	Non-Data Select and Test
	(NDSLX	TTR	NDATA	Non-Data Select Without Test
.MWR	(PROUT	TTR	PROUT0	Message Writer
.PUNCH	(PUNCH	TTR	PUNCH0	Alphameric Punch
.ENBSW	(ENBSW	HTR	**	Enable Switch
.PAWS	(PAWSX	TTR	PAWS	Error Pause
.PAUSE	(PAUSE	TTR	PAUSE	Operator Action Pause
.STOP	(STOPX	TTR	SYSDMP	Post-Mortem Dump
.SYMUN	(SYMUN	TTR	SYUNCV	Symbolic Unit Conversion
.DECVD	(DECVD	TTR	BCVDEC-1	Binary to Decimal - AC Decrement
.DECVA	(DECVA	TTR	BCVDEC	Binary to Decimal - AC Address
.CKWAT	(CKWAT	TTR	CKWAIT	Checkpoint Wait
.BCD5R	(BCD5R	TTR	BCD5-1	Binary to BCD Octal, Bits 3-17 of MQ
.BCD5X	(BCD5X	TTR	BCD5	Binary to BCD Octal, Bits 1-14 and S of MQ
.CVPRT	(CVPRT	TTR	CVPRT	Convert and Append Unit Designation to Message
.STOPD	(STOPD	TTR	SYSDMP	Post-Mortem Dump
.CHXAC	(CHXAC	PZE	CHXAC,1	Channel Activity (Indirect Reference)
.URRX	(URRXI	PZE	URRX,1	Redundancy Count (Indirect Reference)
.RCTX	(RCTXI	PZE	RCTX,1	Redundancy Control (Indirect Reference)
.RCHX	(RCHXI	PZE	RCHX,1	Reset Load Channel (Indirect Reference)
.TCOX	(TCOXI	PZE	TCOX,1	Channel Delay (Indirect Reference)
.TRCX	(TRCXI	PZE	TRCX,1	Tape Redundancy Test (Indirect Reference)
.ETTX	(ETTXI	PZE	ETTX,1	End Tape Test (Indirect Reference)
.TEFX	(TEFXI	PZE	TEFX,1	End File Test (Indirect Reference)
.TRAPX	(TRAPX	PZE	(TRAPS	Current Traps Enabled (Indirect Reference)
.TRAPS	(TRAPS	OCT	377	Current Traps Enabled
.COMM	(COMMM	PZE	**,,**	Store Channel Results at Trap
.LTPOS	(LTPOS	PZE	**,,**	Tape Position Before Last Trap
.IOXSI	(IOXSI	BSS	1	Sense Indicators at Trap
.CHPSW	(CHPSW	PZE	**	Checkpoint Switch
.TRPSW	(TRPSW	PZE	**	Trap Switch
.FDAMT	(FDAMT	TTR	FDAMT	Form Disk/Drum Order
.SDCXI	(SDCXI	PZE	SDCX,1	SDCX Table (Indirect Reference)
.STCXI	(STCXI	PZE	STCX,1	STCX Table (Indirect Reference)
.COMMD	(COMMD	PZE	**	Store Channel Diagnostic Results
.IBCDZ	(IBCDZ	CRQ	DECRQ,,6	BCD Zero Conversion
.CHXSP	(CHXSP	PZE	CHXSP,1	Priority Switch Table

Figure 17. IOEX Communication Table

Function

The function of the System Editor is to modify, add, delete, replace, or rearrange records of the IBSYS Operating System in order to meet the requirements of a particular installation. The IBSYS Operating System is located on one or more System Library Units. The System Monitor (IBSYS) must reside on SYSLB1. The remaining subsystems and the System Editor may also reside on SYSLB1 or may, if so desired, be split to reside on SYSLB2, SYSLB3, and/or SYSLB4. The four system library functions may be assigned any combination of input/output devices (729 Magnetic Tape, 7340 Hypertape, or direct access storage). However, if a subsystem resides on direct access storage, the System Monitor (IBSYS) must also reside on direct access storage. Editing may proceed from any type of input/output unit (729, 7340, 1301, 2302, or 7320) to the same or any other type of unit.

The maximum record size that can be processed by the System Editor is 23,840-word records.

When either a $IBEDT card or a $EXECUTE EDITOR card is read by the System Supervisor, the System Editor is called into core storage and control is relinquished to it. The System Editor then proceeds to read and process control cards on the system input file.

When a new System Library is produced by the Editor, it is recorded on SYSUT1. It may be formed from a combination of input from the old System Library on SYSLB1 or SYSLB2, from alteration cards on SYSIN1 and/or SYSUT2, or from records on any other specified System Unit.

*EDIT Card

Control information is transmitted to the System Editor by an *EDIT card. This control card is required for every edit run and must immediately follow the $IBEDT card or the $EXECUTE EDITOR card, whichever is used, whether or not any options are specified on the control card. The format of the *EDIT card is:

7	16
*EDIT	[SYSLB2][{,HIGH / ,LOW}][,MAP]
	[,MODS][{,xDam/s / ,xNam/s / ,xHk/s}]

The options on this control card are interpreted as follows:

SYSLB2	This option specifies that SYSLB2 is to be edited. If no option is indicated, SYSLB1 is edited.
HIGH LOW	This option specifies the density setting of the new System Library Tape (on SYSUT1). If no density is specified, the density is the same as the old System Library Tape.
MAP	This option specifies that the names of the records and files on the new System Library be listed on the System Output Unit.
MODS	This option specifies that the maintenance control cards (*MODIFY, *REPLACE, etc.) or OCTAL alteration cards that affected a record be listed before the record name on the list specified by the MAP option. If the MAP option is not specified, this option is nullified.
xDam/s xNam/s xHk/s	If the IBSYS record is being edited to a new System Library on disk, drum, or Hypertape, a load card is required; that is, a card is required which loads in the IBSYS record at initial start when it is placed in the card reader and the LOAD CARDS push button on the Operator's Console is pressed. When this option is present, a load card is punched out on the System Punch Unit at the end of the edit. The card loads the IBSYS record from the unit specified by xDam/s (disk) ,xNam/s (drum),or xHk/s (Hypertape). No card is punched if no unit is specified.

Any text in columns 55 through 72 of the *EDIT card is used as the heading of all printed output from the System Editor.

Arrangement of Subsystems

The arrangement of subsystems in the System Library is indicated in the System Name Table (SYSNAM) of the System Supervisor. The table is used by the System Supervisor to locate a subsystem when a $EXECUTE card is read from the input file. There are two entries in the table for each subsystem, as follows:

BCI	1, sysnam
PZE	tfiles, index, nfiles

In the first entry, sysnam is the name of the subsystem. It corresponds to the name on the $EXECUTE card that calls the subsystem. In the second entry, tfiles is the number of consecutive files that make up the subsystem; index is the number 1, 2, 3, or 4, corresponding to SYSLB1, 2, 3, or 4, respectively; and nfiles is the number of files the System Supervisor must skip before scatter-loading the first record of the subsystem. The System Monitor (IBSYS) file is always the first file on SYSLB1, and the System Editor is normally the last file on SYSLB1.

*PLACE Card

The *PLACE card is used to modify the System Name Table. It is the only control card that can cause a change in the System Name Table.

NOTE: All *PLACE cards must immediately follow the *EDIT card.

The format of the *PLACE card is:

7	16
*PLACE	sysnam [, tfiles, index [, order]]

The *PLACE card causes a subsystem name and data concerning the location of the subsystem to be inserted into or deleted from the System Name Table.

If the three arguments tfiles, index, and order are specified on the *PLACE card, an entry is posted in the System Name Table for the subsystem specified by sysnam. The argument tfiles is the number of consecutive files that make up the subsystem; index is the number 1, 2, 3, or 4, corresponding to SYSLB1, 2, 3, or 4, respectively; and order is the order of the specified subsystem in the System Library Unit with respect to other subsystems. The IBSYS file is not a subsystem, so an order of 1 is used to specify the first subsystem, regardless of whether the index is 1, 2, 3, or 4.

If the argument order is omitted or zero and the arguments tfiles and index are included on the control card, the entry in the System Name Table will indicate that the specified subsystem is the last subsystem on the new System Library Unit.

If the three arguments tfiles, index, and order are omitted from the *PLACE card, the entry in the System Name Table for the specified subsystem is deleted.

When editing to direct access storage, a System Loader Table (SLTABLE), as well as the System Name Table, is maintained by the System Editor. When the System Library is on direct access storage, the System Supervisor refers to the System Loader Table, instead of the System Name Table, to locate a subsystem when a $EXECUTE card is encountered. However, when editing to direct access storage, the System Name Table must still be kept up to date, using the *PLACE card. When editing from direct access storage to tape, the new System Library is generated in accordance with the System Name Table on the direct access storage and the System Loader Table is omitted.

When the System Library is on direct access storage, the *PLACE card may also be used to insert or delete home address (HA2) identifiers. This use of the *PLACE card is described in the section "Disk and/or Drum Editing."

Maintenance of the System Name and Loader Tables

The System Name Table is used by the System Supervisor and is, therefore, part of the IBSYS record. When the System Supervisor is loaded into core storage, the System Name Table is loaded into locations SYSORG through SYSORG+99. When the System Editor is subsequently loaded into core storage, it is loaded so that it does not overlay the System Name Table. Therefore, the table is available for updating by the System Editor if any *PLACE cards are read. When the IBSYS record, which is the first record on tape, is edited onto SYSUT1, the System Editor performs the following steps:

1. Examines the location SYSCOR in the Communication Region of the System Nucleus (refer to Appendix A) to determine the current location of the System Name Table.
2. Changes the table as specified by any *PLACE cards.
3. Transfers the updated table to the IBSYS record that is going to be written on SYSUT1. The table is placed in the new IBSYS record, so that it will be located beginning at SYSORG whenever the System Supervisor is loaded into core storage.

When editing to direct access storage, the System Loader Table is automatically maintained by the System Editor. The System Loader Table is located at SYSORG+107 through SYSORG+946. The System Name Table is always maintained by the System Editor from information supplied by *PLACE cards, regardless of whether the System Library is on disk, drum, or tape.

Alteration Cards

Standard System Library records are derived from two types of alteration cards: absolute column-binary and octal.

Absolute Column-Binary Cards

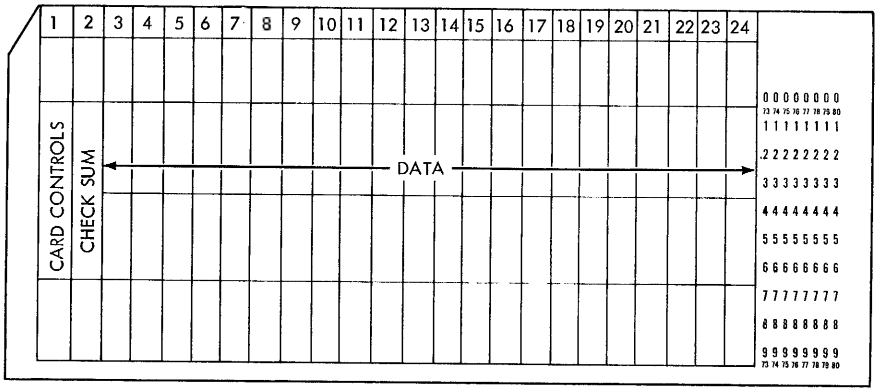
The absolute column-binary cards are the standard 22-word-entry type, illustrated in Figure 18. The data or instruction words, beginning with word 3 on the control card, are written on the System Library Unit in a standard System Library record format that enables them to be loaded in sequential core-storage locations, beginning with the location specified in bit positions 21 ... 30 of word 1.

Octal Cards

The octal card format is as follows:

```
1          7          16                              72
octloc     {*OCT  }   word 1, word 2,...,word n
           { OCTAL}
```

The use of * in column 7 is optional. Each octal word must consist of 12 or fewer unsigned digits. When fewer than 12 digits are entered in a word, they are right-justified in the word. The octal words are separated by commas on the control card. The words will be written on the System Library Unit in a standard System Library record format that enables them to be loaded into sequential core-storage locations, beginning with the location specified by the octal address octloc.



Word	Bit Positions	Contents
1	S	Must be blank.
	1	Must be blank.
	2	If punched, the System Editor does not compute a check sum from card data for comparison with card's prepunched check sum.
	9	Must be punched.
	11	Must be punched.
	12-17	Count of words in the card, excluding words 1 and 2.
	21-35	Absolute loading address -- the actual address of the core storage location where the first data or instruction word on this card is to be stored.
2	S-35	Check sum -- the logical sum (Add and Carry Logical Word) of all words in this card except word 2.
	3-24	Data or instruction words.

Figure 18. Absolute Column-Binary Card Format

Standard System Library Record Formats

Before records are edited onto 729 tape, they are converted by the System Editor into a standard self-loading scatter-load format (shown in Figure 19) that enables them to be loaded by the System Loader in the Nucleus. The records may be converted from similar standard System Library records on disk, drum or Hypertape (shown in Figure 20), from column-binary alteration cards or card images, from octal alteration cards or card images, or from a combination of these.

The System Loader uses the following input/output command sequence to load standard System Library records from 729 tape:

```
IOCP    *+1,,1
xxxx    **,,**
TCH     *-2
```

where xxxx is either an IOCP or IOCT command.

Since the System Supervisor employs the System Loader to load the first record of a subsystem, it is mandatory that the first record be in the standard System Library format shown in Figure 19. A subsystem may also use the System Loader to load succeeding records, provided the records are also in the standard System Library format. No IOCP command used by the System Loader can have a word count greater than $37777)_8$or be in the non-transmitting mode (bit 19 on). No more than one record at a time may be loaded using the System Loader, since redundancy checking is performed and the instruction sequence BSR, RDS is attempted 10 times, if necessary.

Each subsystem consists of one or more files, each file consisting of one or more records. The first record of each subsystem must load the location SYSTRA, in the Communication Region of the Nucleus, with a transfer to the first instruction in the subsystem that is to be executed. This is necessary because the System Supervisor transfers control to SYSTRA after loading the first record of a subsystem specified on the $EXECUTE card.

The first word in a standard record, following the initial input/output command (location LOC1 in Figure 19), must be the name of the record in BCD form, without leading blanks. If a record is the first record of a file, the name of the record is also the name of the file. If the file is the first file of a subsystem, the name of the first record of the file is the name of the subsystem (as specified in the System Name Table), as well as the

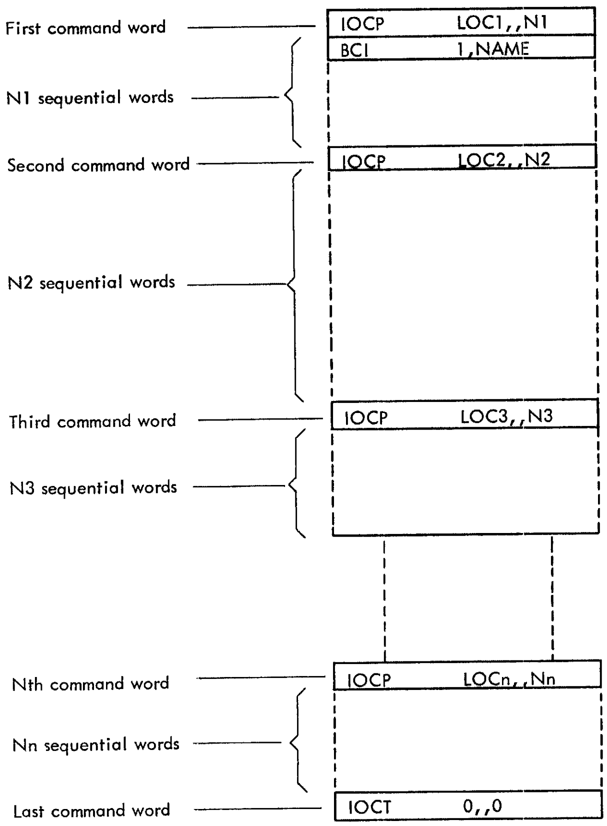
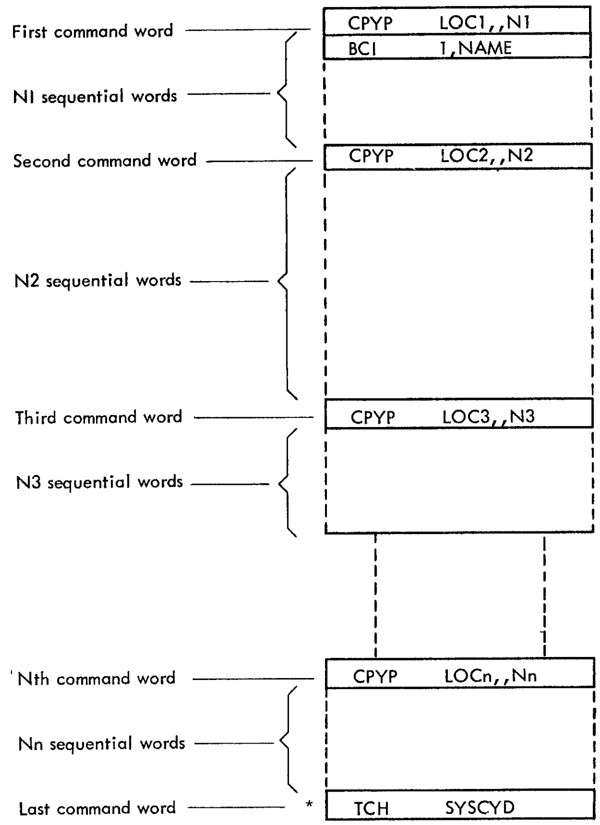


Figure 19. Standard System Library Record Format for 729 Tape



* For an explanation of SYSCYD refer to Appendix A

Figure 20. Standard System Library Record Format for Hypertape, Disk, and Drum

name of the first file of the subsystem. When the System Library is being edited, the name of the first record of each subsystem in the new System Library is checked against the entry for the subsystem in the System Name Table. If a discrepancy exists, an error message is printed.

The record name is the name printed when the MAP option on the *EDIT card is exercised. It is also used by the System Editor to identify a record, a file, or a subsystem.

Note, however, that the System Supervisor locates a subsystem specified on a $EXECUTE card by referring to the System Name Table or, if the System Library is on direct access storage, by referring to the System Loader Table. The last record on a System Library Tape has the identification *EOT and is used to inform the System Editor of the logical end of tape.

When a subsystem or part of a subsystem is loaded into core storage, it must not overlay the System Nucleus.

If the subsystem is not loaded into the area occupied by the Input/Output Executor (IOEX) or the off-line writing routine SPOUT, IOEX and SPOUT are immediately available for use by the subsystem. The off-line writing routine SPOUT cannot be used if IOEX is overlaid.

Before records are edited onto Hypertape or direct access storage, they are converted by the System Editor into a standard System Library format (shown in Figure 20) similar to the format for 729 tape, that enables them to be loaded by the System Loader. The records may be converted from standard System Library records on 729 tape (shown in Figure 19), from column binary alteration cards or card images, from octal alteration cards or card images, or from a combination of these. The System Loader uses the following input/output command sequence to load standard System Library records from Hypertape or direct access storage:

```
CPYP    *+1,,1
xxxx    **,,**
TCH     *-2
```

where xxxx is either a CPYP or TCH command.

Since the System Supervisor employs the System Loader to load the first record of a subsystem, it is mandatory that the first record be in the standard System Library format as shown in Figure 20.

When loading standard System Library records from direct access storage, the cylinder mode of operation is used. A sample scatter-read program for reading records from disk is shown in Appendix E.

When a record on direct access storage extends across two cylinders, the System Editor sets the last word on the first of the two cylinders to the following:

```
TCH        SYSCYD,,yyy
```

where yyy is the location of the first track of the second of the two cylinders. The remainder of the record on the second cylinder begins with an appropriate CPYP command. If editing proceeds from direct access storage to magnetic tape, any TCH command within a record that was written because the record straddled two cylinders is reset by the System Editor to a CPYP command with a 0 word count.

In a disk or drum record, the low-order bits of the decrement in the final TCH SYSCYD command points to the track origin of the next sequential record, and the high-order bit of the decrement is set to 1. The 1-bit indicates to the System Loader that the TCH is the true end of the record and not just the end of a cylinder.

Maintenance Control Cards

The actual addition, deletion, rearrangement, or modification of subsystems or records and files within a subsystem is performed by the use of the maintenance control cards described in this section. A *PLACE card does no more than post, in the System Name Table, the position of a subsystem. The maintenance control cards must be in the same order as the records to which they refer. If a maintenance control card refers to a record out of sequence in the edit deck, the effect of the control card is nullified and a message is printed.

*MODIFY Card

The format of the *MODIFY card is as follows:

```
1            7          16
[TAPE]       *MODIFY    recnam
```

where recnam is the name of a record in the old System Library (SYSLB1 or SYSLB2) that is to be modified.

If TAPE does not appear in columns 1-4, this control card causes the specified record to be consolidated with alteration cards that follow this control card on SYSIN1. The alteration cards on SYSIN1 may be any combination of octal and column-binary cards.

If TAPE appears in columns 1-4, the *MODIFY card causes the specified record to be consolidated with alteration cards on SYSUT2. The alteration cards on SYSUT2 must be in the form of column-binary card images.

The System Editor performs the following steps after reading a *MODIFY card:

1. Transfers to SYSUT1 all files, file marks, and records on SYSLB1 (or SYSLB2, if specified on the *EDIT card) up to, but not including, the record specified on the *MODIFY card. If necessary, records are converted to the standard System Library record format shown in Figure 19, when SYSUT1 is 729 tape, or to the standard format shown in Figure 20, if SYSUT1 is Hypertape or direct access storage.
2. Reads the specified record into core storage.
3. Deletes the IOCT or TCH command at the end of the record.
4. Appends an IOCP (or CPYP) command of the following form to the record:

   ```
   IOCP        LOC,,N
   ```

   where LOC is the storage location specified in the first alteration card and N is the number of words, contained on the first alteration card and the alteration cards immediately following it, that are to be loaded into contiguous core-storage locations.
5. Places the N words from the alteration cards into the record following the newly appended command word.
6. Repeat steps 4 and 5, with appropriate LOC and N values, until, if TAPE is specified, a transfer card is encountered on SYSUT2 or, if TAPE is not specified, until a new System Editor control card is encountered on SYSIN1.
7. Appends, to the end of the record, an IOCT command with a word count of zero or, if SYSUT1 is Hypertape or direct access storage, a TCH SYSCYD command.
8. Writes the expanded record onto SYSUT1.

After all actions required by the *MODIFY card are completed, SYSLB1 (SYSLB2) will be positioned just after the specified record and before the next record (or file mark, if not a record).

*REPLACE Card

The format of the *REPLACE card is as follows:

```
  1          7          16
[ {FILE} ]   *REPLACE {recnam} , SYSxxx
[ {TAPE} ]            {sysnam}
```

If neither TAPE nor FILE appears in columns 1-4, the *REPLACE card causes an entire record (specified by recnam) in the old System Library on SYSLB1 (SYSLB2) to be replaced on SYSUT1, with a new record formed entirely from the alteration cards that follow the *REPLACE card on SYSIN1. If TAPE appears in columns 1-4, the specified record is replaced by a new record formed from the alteration cards on SYSUT2, rather than SYSIN1.

Before the specified record is replaced, all of the files, file marks, and records that precede the record on SYSLB1 (SYSLB2) are transferred to SYSUT1. The replacement record represented by the alteration cards and, if necessary, the records transferred from SYSLB1 (SYSLB2) are converted to the standard System Library record format shown in Figure 19 if SYSUT1 is 729 tape,

or to the standard format shown in Figure 20 if SYSUT1 is Hypertape or direct access storage.

The alteration cards on SYSUT2 must be column-binary card images when TAPE is specified. When TAPE and FILE are not specified, the alteration cards that follow the *REPLACE card on SYSIN1 may be any combination of octal and column-binary cards. As with the *MODIFY card, the end of the alteration cards is signaled by a transfer card when TAPE is specified, or by a System Editor control card when TAPE and FILE are not specified.

When all of the operations called for by the *REPLACE or TAPE *REPLACE card are completed, SYSLB1 (SYSLB2) is positioned just after the record specified by recnam, but before the next record or file mark.

If FILE appears in columns 1-4 of the *REPLACE card and the name appearing in columns 16-21 matches a subsystem name in the System Name Table, all records, files, and file marks are read in from SYSLB1 (SYSLB2) and written out on SYSUT1, until the specified subsystem is located in accordance with the entry in the System Name Table. When the subsystem is located, it is replaced on SYSUT1 by the files on SYSXXX. If the number of files in the replacement differs from the original, a *PLACE card reflecting this must be inserted into the edit deck following the *EDIT card, but preceding the maintenance control cards.

The first word of each record read from SYSXXX is checked for an IOCP (or CPYP) command. If it is not an IOCP (or CPYP) command, the record is considered nonstandard and is duplicated without change onto SYSUT1. If it is an IOCP (or CPYP) command, the record is considered a standard System Library record. In this case, the record is duplicated onto SYSUT1, except for the last word. The last word is changed to an IOCT 0,,0 command if SYSUT1 is 729 tape, or to a TCH SYSCYD command if SYSUT1 is Hypertape or direct access storage. If SYSUT2 is a direct access storage unit, the required TCH command is inserted whenever a cylinder boundary is crossed while a record is being written.

If FILE appears in columns 1-4 of the *REPLACE card and the name appearing in columns 16-21 does not match a subsystem name in the System Name Table, all records, files, and file marks are read in from SYSLB1 (SYSLB2) and written out on SYSUT1, until a record with the specified name is located. When the specified record is located, it, and all succeeding records (if any) up to and including the next file mark, are replaced on SYSUT1 by the next file and file mark on SYSXXX. The replacement records are processed by the System Editor as described in the preceding paragraph. To replace an entire file, the record name specified on the FILE *REPLACE card must be the name of the first record in the file. However, if the name of the first record in the file matches a subsystem name in the System Name Table, all of the files in the subsystem will be replaced, as previously described.

When all actions required by the FILE *REPLACE card are completed, SYSLB1 (SYSLB2) is positioned after the file mark that follows the last file (or part of a file) replaced.

**INSERT Card*

The format of the *INSERT card is as follows:

```
1          7          16
[TAPE]     *INSERT    [FILEMK]
```

If TAPE and FILEMK are not specified, the *INSERT card causes a new record, formed from octal and/or column-binary alteration cards following the *INSERT card on SYSIN1, to be written on SYSUT1 at its current position. The end of the alteration cards on SYSIN1 is indicated by a new System Editor control card.

If TAPE appears in columns 1-4 and FILEMK is not specified, the new record is formed from alteration cards in the form of column-binary card images on SYSUT2. The end of the alteration cards on SYSUT2 is indicated by a transfer card.

The new record, represented by the alteration cards on SYSIN1 or SYSUT2, is converted to the standard System Library record format shown in Figure 19, if SYSUT1 is 729 tape, or to the standard format shown in Figure 20, if SYSUT1 is Hypertape or direct access storage.

If FILEMK is specified, a file mark is inserted on SYSUT1 at its current position. When SYSUT1 is disk or drum, a flag, analogous to a tape file mark, is placed into a disk/drum record address area (shown in Figure 21). A single *INSERT card cannot be used both to insert a new record and to write a file mark.

The position of SYSLB1 (SYSLB2) is not changed by an *INSERT card.

**REMOVE Card*

The format of the *REMOVE card is as follows:

```
1          7          16
[FILE]     *REMOVE   {recnam
                      sysnam
                      FILEMK}
```

If neither FILE nor FILEMK is specified, the *REMOVE card causes the record specified in columns 16-21 to be spaced over on SYSLB1 (SYSLB2) and omitted from SYSUT1. Before the specified record is removed, all of the files, file marks, and records that precede the record on SYSLB1 (SYSLB2) are transferred to SYSUT1. After the *REMOVE card is processed, SYSLB1 (SYSLB2) is positioned immediately after the specified record.

If FILEMK is specified in columns 16-21, the next end-of-file mark encountered on SYSLB1 (SYSLB2) is spaced over and omitted from SYSUT1. All records that precede the file mark on SYSLB1 (SYSLB2) are transferred to SYSUT1. SYSLB1 (SYSLB2) ends up positioned immediately after the file mark that was omitted from SYSUT1.

If FILE is specified and the name in columns 16-21 is a subsystem name in the System Name Table, all the files and accompanying file marks associated with the subsystem (as defined by the System Name Table entry) are spaced over on SYSLB1 (SYSLB2) and omitted from SYSUT1.

If FILE is specified and the name in columns 16-21 is the name of a record, but not the name of a subsystem posted in the System Name Table, the specified record and all succeeding records (if any), up to and including the next file mark, are spaced over on SYSLB1 (SYSLB2) and omitted from SYSUT1. If an entire file is to be removed, the name specified on the FILE *REMOVE card must be the name of the first record in the file. However, if the name of the first record in the file matches a subsystem name in the System Name Table, all of the files in the subsystem will be removed, as previously described.

Before a specified subsystem or record is removed, all of the records, files, and file marks that precede it on SYSLB1 (SYSLB2) are transferred to SYSUT1.

*AFTER Card

The format of the *AFTER card is as follows:

```
1          7          16
[FILE]     *AFTER    ⎧recnam ⎫
                     ⎨sysnam ⎬
                     ⎩FILEMK ⎭
```

If neither FILE nor FILEMK is specified, the *AFTER card causes the reading of control cards to be suspended until all files, file marks, and records on SYSLB1 (SYSLB2), up to and including the record specified in columns 16-21, have been transferred to SYSUT1.

If FILEMK is specified in columns 16-21, the next file mark on SYSLB1 (SYSLB2) and all records preceding it are transferred to SYSUT1.

If FILE is specified and the name in columns 16-21 is a subsystem name in the System Name Table, all of the files, file marks, and records on SYSLB1 (SYSLB2), up to and including the files and accompanying file marks associated with the specified subsystem (as defined by the System Name Table entry), are transferred to SYSUT1.

If FILE is specified and the name in columns 16-21 is the name of a record, but not the name of a subsystem posted in the System Name Table, all of the f.'es, file marks, and records on SYSLB1 (SYSLB2), up to and including the next file mark following the specified record, are transferred to SYSUT1.

*DUP Card

The format of the *DUP card is as follows:

```
7          16
*DUP       SYSxxx, SYSyyy, n
```

This control card transfers n files from SYSXXX to SYSYYY. The transfer proceeds up to and through the nth file mark read on SYSXXX. If n is blank, 1, or 0, one file and the file mark following it are transferred to SYSYYY.

If SYSYYY is SYSUT1, the transfer of files from SYSXXX is performed as described for the FILE *REPLACE card, as follows. The first word of each record of the files read from SYSXXX is checked for an IOCP (or CPYP) command. If it is not an IOCP (or CPYP) command, the record is considered nonstandard and is duplicated without change onto SYSUT1. If it is an IOCP (or CPYP) command, the record is considered a standard System Library record. In this case, the record is duplicated onto SYSUT1, except for the last word. The last word is changed to an IOCT 0,,0 command if SYSUT1 is 729 magnetic tape, or to a TCH SYSCYD command if SYSUT1 is Hypertape or direct access storage. If SYSUT1 is direct access storage, the required TCH command is inserted whenever a cylinder boundary is crossed while a record is being written.

*REWIND Card

The format of the *REWIND card is as follows:

```
7          16
*REWIND    SYSxxx
```

This control card rewinds the unit assigned to the specified system unit function. When the unit assigned to the specified system unit function is a direct access storage unit, the *REWIND card causes the System Editor to begin its next read or write operation on the unit at the load point defined by the parameters on the $AS card when the unit was assigned to the specified system unit function.

The *REWIND card may be used to define the starting tape position or disk/drum load point before using a *DUP card. The *REWIND and *DUP cards may be used to rearrange subsystems in the System Library or to incorporate, into the System Library, special-format records that have been previously edited by a subsystem.

*CHECK Card

The format of the *CHECK card is as follows:

```
7          16
*CHECK     count, oldnam, newnam
```

This control card causes a test to be made to ensure that the correct number of editing cards were read and that the correct System Library tape was processed. The argument count is the number of alteration cards and maintenance control cards, including the *CHECK card, but not the *EDIT card. The argument oldnam is a name which is compared with the System Library name in the fourth word of the *EOT record on SYSLB1 (SYSLB2). If the two names are not the same, an error message is printed. The argument newnam is the name assigned to the new System Library (tape, disk, or drum) on SYSUT1. The arguments must appear

in the order given. If an argument is blank, the corresponding operation is ignored.

If more than one *CHECK card is read during an edit run, only the last one is processed.

REMARK Card

The format of the *REMARK card is as follows:

```
7           16
*REMARK     any remark
```

This control card causes the characters in columns 16-72 of the control card to be listed on the System Printer and the System Output Unit.

The *REMARK card may not be placed between the control cards *MODIFY, *REPLACE, or *INSERT, and the alteration cards immediately following these control cards.

Termination of Editing

The System Editor completes the editing process when an end of file or a control card with a 7 and 8 punch in column 1 is encountered on SYSIN1. The System Editor performs the following steps to terminate the editing process:

1. Transfers from SYSLB1 (SYSLB2) to SYSUT1 all remaining files, file marks, and records up to, but not including, the *EOT record on SYSLB1 (SYSLB2).
2. Reads the *EOT record from SYSLB1 (SYSLB2) and performs the tests and changes specified on the last *CHECK card read.
3. Writes the new *EOT record and a file mark on SYSUT1.
4. Rewinds SYSLB1 (SYSLB2) and SYSUT1.

The System Editor then reads the next card on SYSIN1, expecting to find a $IBSYS, $JOB, $EXECUTE, or $STOP card.

Disk and/or Drum Editing

When editing to direct access storage, a System Loader Table, as well as the System Name Table, is maintained by the System Editor. When the System Library is on disk and/or drum, the System Supervisor refers to the System Loader Table, instead of the System Name Table, to locate a subsystem when a $EXECUTE card is encountered. However, when editing to direct access storage the System Name Table must still be kept up to date, using the *PLACE card. The System Loader Table is part of the IBSYS record which must be the first record edited to direct access storage.

Records which do not have an IOCP or CPYP command as their first word, containing the number of words following, are not included in the System Loader Table. A record is not considered to be a standard System Library record if it begins with any of the following commands: IORT, IORP, IOCT, IOSP, or IOST.

An end of record or an end of file is indicated on direct access storage by a flag in the record address

E O R	E O F		Seek Chain Address		"Pseudo" Record Count to simulate SCHx Instruction
S	1	2	3 — 17	— 21	— 35

Figure 21. Format of Disk/Drum Record Address Area

area. The format of a record address area is shown in Figure 21. An end of record is indicated by a 1-bit in bit position S and an end of file is indicated by a 1-bit in bit position 1.

The System Editor writes using the home address operation. Disk writing is done in the 6-bit mode using a 465-word track format for 1301 disk and a 969-word format track for 2302 disk. A 524-word track format is used when writing on drum. Any 1301/2302 Disk Storage or 7320 Drum Storage used by the System Editor must have the Verify Cylinder feature.

If an error occurs while editing to direct access storage, the format track, which was assumed good at the start, is rewritten to ensure its validity, and editing continues. If a second error occurs on the same select request, editing continues, if possible, and the condition is ignored.

When editing involves a direct access device, the HA0 switch on the IBM 7631 File Control must be on (UP). To protect the System Library, a home address (HA2) identifier containing BM0000 is written on each track containing records of the System Library. The format key should be unlocked to enable the System Editor to rewrite format tracks. It should be locked again when the edit run is completed.

The *PLACE card may be used to insert or delete home address (HA2) identifiers. This use of the *PLACE card is illustrated by the following examples.

Example 1

Starting with cylinder 10, write home address (HA2) identifiers of PQ on 15 cylinders assigned to the specified system unit function.

```
1         7         16
SYSxxx    *PLACE    PQ, 15, 10
```

Example 2

Reset to general use code 00 the home address (HA2) identifiers on 250 cylinders assigned to the specified system unit function beginning with cylinder 0.

```
1         7         16
SYSxxx    *PLACE    00, 250, 0
```

Care should be taken to ensure that the cylinder limits are defined correctly, since incorrectly defined limits may result in the destruction of valid data.

Example 3

If it is desired to write home address (HA2) identifiers on the cylinders assigned to a specified system unit function and to bypass subsequent editing, the follow-

ing *PLACE card should be inserted in the edit deck, after all other *PLACE cards and preceding the end-of-file card.

```
1          7
SYSxxx     *PLACE
```

Editing Relocatable Records

If a standard relocatable record is located on SYSUT2 when one of the following maintenance control cards is read, the System Editor converts it to a format similar to the standard System Library format.

```
1          7         16
TAPE       *REPLACE recnam
TAPE       *INSERT  recnam
```

Two IOCP (CPYP) commands and a record name are placed at the beginning of the record by the Editor, as follows:

```
IOCP    LOC1,,1
BCI     1,NAME
IOCP    LOC2,,N
```

where LOC1 is 45000_8, NAME is the record name specified on the *REPLACE or *INSERT card, LOC2 is 45002_8, and N is the number of words in the relocatable record. An IOCT command with a word count of zero is appended to the record if SYSUT1 is 729 tape, or a TCH SYSCYD command is appended to the record if SYSUT1 is Hypertape or direct access storage.

Once a relocatable record is placed in the System Library in the standard System Library format, all maintenance control cards that are applicable to non-relocatable records in the standard System Library format, except the *MODIFY card, are also applicable to the relocatable record.

Editing Examples

The following are examples of typical editing jobs that might be performed by the System Editor.

Example 1

The sample job deck in Figure 22 might be used to prepare a new 729 System Library Tape with corrections to record REC1 and complete replacement of records ABC and XYZ.

Example 2

The sample job deck in Figure 23 might be used to prepare, from a 729 System Library Tape, a new System Library on channel C, Hypertape Drive 2, Data Channel Switch setting 2. The new System Library Tape is to be used on channel C, Hypertape Drive 0, Data Channel Switch setting 2.

Example 3

The sample job deck in Figure 24 might be used to prepare, from the System Library Tape prepared in editing example 2, a new System Library on channel C, disk module 1, Data Channel Switch setting 1.

Example 4

The sample job deck in Figure 25 might be used to rearrange the subsystems on a 729 System Library Tape so that the subsystem SYSTMA, consisting of two files located just after the IBSYS file, will be repositioned just after the subsystem SYSTMX. Assume SYSTMX is presently the last three files just before the System Editor.

```
1     7        16
------------------------------------------------------------
$JOB           729 TAPE TO 729 TAPE EDIT
$IBSYS
$IBEDT
      *EDIT    MAP,MODS
      *MODIFY  REC1
      .
      .
      .
      .   (*OCT AND/OR COLUMN-BINARY
      .      CARDS ON SYSIN1)
      .
      .
      .
TAPE  *REPLACE ABC
   (COLUMN-BINARY CARD IMAGES ON SYSUT2)
TAPE  *REPLACE XYZ
   (COLUMN-BINARY CARD IMAGES ON SYSUT2, SEPARATED
      FROM ABC CARD IMAGES BY A TRANSFER CARD)
   (END-OF-FILE CARD)
$STOP
```

Figure 22. Sample Job Deck for Making Corrections to a 729 System Library Tape

```
1     7        16
------------------------------------------------------------
$JOB           729 TAPE TO HYPER EDIT
$IBSYS
$ATTACH        CH2/1
$AS            SYSUT1
$IBEDT
      *EDIT    MAP,MODS,CH0/1
      *CHECK   1,729SYS,HYPSYS
   (END-OF-FILE CARD)
$IBSYS
```

Figure 23. Sample Job Deck for Preparing a System Library on Hypertape from a 729 System Library Tape

```
1     7        16
------------------------------------------------------------

    LOAD USING HYPERTAPE LOAD CARD PUNCHED
      DURING JOB LISTED IN FIGURE 23

$JOB           HYPER TO DISK EDIT
$IBSYS
$ATTACH        CD01/0
$AS            SYSUT1,010,222,BM
$IBEDT
      *EDIT    CD01/0
   (END-OF-FILE CARD)
$STOP
```

Figure 24. Sample Job Deck for Preparing a System Library on Disk Storage from a System Library on Hypertape

```
1     7         16
------------------------------------------------------------------------
$JOB            REARRANGE SYSTMA ON 729 SYSTEM LIBRARY TAPE
$IBSYS
$IBEDT
      *EDIT     MAP,MODS
      *PLACE    SYSTMA               DELETE OLD SYSTMA REFERENCE IN SYSNAM
      *PLACE    SYSTMA,2,1           INSERT NEW SYSTMA REFERENCE IN SYSNAM
      *AFTER    FILEMK               POSITION SYSLB1 AFTER THE IBSYS FILE
      *REWIND   SYSUT4               ENSURE SYSUT4 IS REWOUND
      *DUP      SYSLB1,SYSUT4,2      DUPLICATE SYSTMA (2 FILES)
FILE  *AFTER    SYSTMX               POSITION SYSLB1 AFTER SYSTMX
      *REWIND   SYSUT4               POSITION SYSUT4 AT BEGINNING OF SYSTMA
      *DUP      SYSUT4,SYSUT1,2      PUT SYSTMA ON NEW LIBRARY TAPE
   (END-OF-FILE CARD)                FINISH UP EDIT
$IBSYS
```

Figure 25. Sample Job Deck for Rearranging Subsystems on a 729 System Library Tape

Creation of an Alternate Library

An alternate library (i.e., a library not containing the System Monitor (IBSYS) file and residing on SYSLB2, 3, or 4) can be created by using the extended *INSERT and *DUP cards described below:

```
1           7          16
[TAPE]      *INSERT    [FILEMK] [,SYSALT]
```

When the SYSALT option is specified, the action called for by the *INSERT card takes place on the unit assigned to SYSUT3 rather than SYSUT1.

```
7          16
*DUP       SYSxxx, SYSALT, n
```

When the SYSALT option is specified, the n files will be transferred from SYSXXX to SYSUT3 and the results on SYSUT3 will be edited in the same manner as they would have been if SYSUT1 had been used.

Example: Assume that a system tape contains, among other things, a subsystem, SYSTMA, consisting of two files following IBSYS. Further assume that a new subsystem, SYSTMB, exists on SYSUT2 in column-binary card image form with one transfer card at the end. The sample job deck shown in Figure 26 could be used to create a new library on SYSLB1, without SYSTMA and SYSTMB, and an alternate library (to be used as SYSLB3) consisting of SYSTMA followed by SYSTMB.

NOTE: The Editor will finish the above edit by placing the *EOT record, which was written on SYSUT1, on SYSUT3 since the SYSALT option was specified at least once during the edit.

Maintenance of an Alternate Library

An alternate library (one without IBSYS) is modified by having it on SYSLB2 and using the SYSLB2 option on the *EDIT card. The edit will proceed in the normal fashion to SYSUT1. All Editor control cards except the *PLACE card and any cards on which SYSALT is specified are available for modification of the alternate library. However, care must be taken not to change the alternate library so much that it is no longer correctly reflected in the System Name Table of the SYSLB1 IBSYS file, since a new System Name Table cannot be generated with this edit. If SYSLB1 is on direct access storage, a new System Loader Table reflecting the edit (such as new track origins of the systems on the alternate library if SYSUT1 is direct access storage) will automatically be rewritten in the IBSYS file residing on SYSLB1.

Note that the discussion above concerns editing, for maintenance purposes, of an alternate library. If the SYSLB2 option is used on the *EDIT card and the library on SYSLB2 contains the IBSYS file, *all* Editor control cards are available. Editing proceeds from SYSLB2 to SYSUT1, and a new System Name Table, taken from the SYSLB1 IBSYS file and changed according to any *PLACE cards, will be written as part of the IBSYS file on SYSUT1. If SYSUT1 is direct access storage, a System Loader Table is generated by the Editor and written on SYSUT1. In contrast to the alternate library edit, the System Loader Table of the IBSYS file on SYSLB1, if it is direct access storage, is not changed.

```
1     7         16
------------------------------------------------------------------------
$JOB
$IBSYS
$IBEDT
      *EDIT     MAP,MODS
      *PLACE    SYSTMA               DELETE SYSTMA REFERENCE IN SYSNAM
      *PLACE    SYSTMA,2,3,1         MAKE NEW ENTRY WITH INDEX 3
      *PLACE    SYSTMB,1,3           INSERT A SYSTMB REFERENCE
      *AFTER    FILEMK               EDIT TO SYSUT1 UP TO SYSTMA
      *DUP      SYSLB1,SYSALT,2      PUT SYSTMA ON SYSUT3
TAPE  *INSERT   SYSALT               FOLLOW WITH SYSTMB ON SYSUT3
      *INSERT   FILEMK,SYSALT        PUT FILE MARK ON SYSUT3
   (END-OF-FILE CARD)                FINISH EDIT ON SYSUT1
$IBSYS
```

Figure 26. Sample Job Deck for Creating an Alternate Library

Introduction

The complete 7090/7094 IBSYS Operating System is preassembled for use with 729 Magnetic Tape Units and is distributed in binary form on a single 729 System Library Tape. This tape can be used immediately without change by any installation employing 729 Magnetic Tape Units, either for production job processing or for preparing a new System Library that meets the specific requirements of an installation.

In addition to the preassembled 729-capability System Library Tape, symbolic tapes containing the complete Operating System are also available. On these tapes, the System Monitor, the System Editor, the IBJOB Processor, the Symbolic Update Program, the Utilities, and the Restart Program are written in the Macro Assembly Program (MAP) language, and the remainder of the Operating System is written in the FORTRAN II Assembly Program (FAP) language. The symbolic tapes may be used, if necessary, to reassemble all or parts of the Operating System. The System Monitor and each of the subsystems on the symbolic tapes contain assembly parameters which may be replaced before assembly to change certain operating characteristics of the System and thereby tailor it to the needs of a particular installation.

If an installation is to employ 1301/2302 Disk Storage and/or 7340 Hypertape Drives, parts of the Operating System on one of the symbolic tapes must be updated (to replace assembly parameters), assembled, and then edited onto a new System Library Tape, together with the remainder of the System. The preassembled 729-capability magnetic tape and an update-edit job deck that is distributed with the System may be used to perform these operations.

The procedure for adding 7320 Drum Storage capability to an installation is similar to the procedure for adding disk and/or Hypertape capability. The exception is that the tape which is produced as a result of running the Disk/Drum/Hypertape Update-Edit Deck must be used in conjunction with the distributed symbolic tape which contains the IBJOB Loader (IBLDR) and the symbolic modification deck shown in Figure 58. This procedure modifies the loader portion of the IBJOB processor for drum storage. (The 7320 Update-Edit Deck Number 2 is available upon request from the DP Program Information Department.) This procedure produces a system tape on which the blocking factor for the IBJOB Subroutine Library (IBLIB) has been changed to utilize full tracks of 7320 Drum Storage. These modifications should be made only if the IBJOB Subroutine Library is to reside on drum storage. The System Library Tape produced by this run is then used in place of the tape obtained by means of the update-edit deck.

This section of the publication contains information and procedures for performing the following:

1. Preparing a backup System Library Tape.
2. Changing System Unit assignments by patching.
3. Preparing duplicate 729 System Library tapes for alternate use by the System Monitor.
4. Adding 1301/2302 Disk Storage, 7320 Drum Storage, and/or 7340 Hypertape capability.
5. Reassembling the System Monitor.
6. Incorporating a user-designed installation accounting routine.
7. Incorporating user programs as subsystems under System Monitor control.
8. Incorporating IBM modifications to the 7090/7094 IBSYS Operating System.

Additional information on changing the operating characteristics of the subsystems operating under System Monitor control is contained in the manuals for the subsystems.

Distributed Configuration of Input/Output Units

The distributed 729-capability System Library Tape is assembled for the configuration of referable input/output units shown in Figure 27. This configuration is designed to accommodate the actual configuration of 729 Magnetic Tape Units existing at most installations. A unit control block is generated by the System Monitor for each of the referable units. However, an actual physical unit need not, and normally does not, exist at a particular installation for each unit control block generated when the distributed tape is used.

Channel	Unit Type	Number	Attached Units	Detached Units
Channel A	711 Card Reader	1	RDA	
	721 Card Punch	1	PUA	
	716 Printer	1	PRA	
	729 Magnetic Tape	10	A1...A8	A9,A0
Channel B	729 Magnetic Tape	10	B1...B8	B9,B0
Channel C	729 Magnetic Tape	6		C1...C6
Channel D	729 Magnetic Tape	6		D1...D6

Figure 27. Configuration of Referable Input/Output Units on Distributed System Library Tape

Unit	System Unit Assignments	Density Assignment
RDA	SYSCRD	
PUA	SYSPCH	
PRA	SYSPRT	
A1	SYSLB1	high
A2	SYSIN1 and SYSIN2	high
A3	SYSUT1	high
A4	SYSUT3	high
A5	SYSCK2	high
B1	SYSOU1 and SYSOU2	high
B2	SYSPP1	high
B3	SYSUT2	high
B4	SYSUT4 and SYSPP2	high

Figure 28. System Unit Assignments on Distributed System Library Tape

On the distributed preassembled System Library Tape, the card reader, the card punch, the printer, and nine of the 729 Magnetic Tape Units are assigned to the system unit functions shown in Figure 28. The remaining system unit functions have no units assigned. In general, the procedures described in this section are based on the assumption that the units listed in Figure 28 are physically connected.

If the System Monitor is assembled from the distributed symbolic tape without prior changes in assembly parameters, it will have the same configuration of referable units (Figure 27) and system unit assignments (Figure 28) as the distributed preassembled System Library Tape.

Preparing a Backup System Library Tape

Although the following procedure is optional, it is strongly recommended, since it produces a System Library map for future reference and a duplicate System Library Tape for emergency use.

1. Place the sample job deck shown in Figure 29 in the system input file on SYSIN1 (A2).

2. Mount the distributed System Library Tape on SYSLB1 (A1).

3. Follow the initial start procedure described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

A duplicate of the distributed System Library Tape will be produced on SYSUT1 (A3), and a map of the System Library will be produced on SYSOU1 (B1).

```
1     7        16
-----------------------------------------------
$JOB           BACKUP SYSTEM LIBRARY
$IBSYS
$DATE          122564
$IBEDT
      *EDIT    MAP
   (END-OF-FILE CARD)
$STOP
```

Figure 29. Sample Job Deck for Preparing a Backup 729 System Library Tape

Changing System Unit Assignments by Patching

The System Unit Function Table in the System Nucleus is generated at initial start from the 24-word Auxiliary System Unit Function Table in the IBSYS record of the System Monitor. A symbolic listing of the distributed version of the auxiliary table is shown in Figure 38. The table is located in core storage, beginning at 15140_8 (SYSLB1) and ending at 15167_8 (SYSUT4). Note that in the distributed version of the table, the same unit (B4) is assigned to both the SYSPP2 and SYSUT4 functions. This unit is so assigned in the distributed version because SYSPP2 functions as a spill tape for the System Core-Storage Dump Program. Therefore, it must have a unit assigned to it. If SYSPP1 and SYSPP2 are to be used alternately for reel switching purposes, then a different unit must be assigned to either SYSPP2 or SYSUT4. The unit assigned to SYSPP2 cannot be direct access storage.

Temporary changes in the assignment of units to system unit functions can be made using the $ATTACH and $AS cards or the $SWITCH card. However, to permanently change the assignment of a unit to a system unit function, the entry for the function in the Auxiliary System Unit Function Table must be changed. This may be done during an update and reassembly of the System Monitor, as described in the sections "Adding 1301 Disk Storage, 7320 Drum Storage, and/or 7340 Hypertape Capability" and "Reassembling the System Monitor." Alternatively, entries in the table can be changed, without reassembly, by patching them (using octal alteration cards) during an edit run, as described in the following text.

Changing the Assignment of 729 Magnetic Tape Units and Card Equipment

Each entry for 729 Magnetic Tape Units and card equipment in the Auxiliary System Unit Function Table has the following format:

pfx   channel,,unit

where pfx is the density assigned to the system unit function: MZE for high density and PZE for low density; and channel is a number from 1 through 8 (corresponding to channels A through H) which indicates the channel of the unit assigned to the system unit function. For the configuration of input/output units on the distributed tape, channel can range from 1 to 2. The value unit is the number of the 729 Magnetic Tape Unit (1 through 10), card reader (11), card punch (12), or printer (13) assigned to the system unit function. As an example, if reel switching is desired for the system input, output, and peripheral punch functions, the sample job deck in Figure 30 might be used. Assuming that the System Library Tape on SYSLB1 is

```
1     7          16
-------------------------------------------------------
$JOB             CHANGE SYSTEM ASSIGNMENTS
$IBSYS
$IBEDT
      *EDIT      MAP,MODS
      *MODIFY    IBSYS
15150 *OCT       400005000002       B5 AS SYSOU2
15152 *OCT       400006000001       A6 AS SYSIN2
15154 *OCT       400006000002       B6 AS SYSPP2
   (END-OF-FILE CARD)
$STOP
```

Figure 30. Sample Job Deck for Changing System Unit Assignments

the distributed tape, this job deck will assign B5 as SYSOU2, A6 as SYSIN2, and B6 as SYSPP2; all at high density.

Changing the Assignment of 7340 Hypertape Drives

Each entry in the Auxiliary System Unit Function Table for 7340 Hypertape Drives has the following format:

PZE channel,,unit

where channel is a number from 1 through 8 (corresponding to channels A through H) and unit is the full address of the Hypertape drive specified in the decrement portion of the first word of its unit control block, as shown in Figure 31.

For example, to permanently assign HH9/1 (channel H, Hypertape drive 9, Data Channel Switch setting 2) as SYSUT2, the following octal alteration card would be used in a job deck similar to the one shown in Figure 30:

15160 *OCT 030051000010

Changing the Assignment of 1301/2302 Disk Storage and 7320 Drum Storage

An entry in the Auxiliary System Unit Function Table for direct access storage is similar to an entry for Hypertape except that bits 14 through 17 specify a module number (0-9 for disk; 0, 2, 4, 6, or 8 for drum), rather than a unit number, and bits 9 through 11 are 001 for 1301 disk, 111 for 2302 disk, and 011 for drum, instead of 000. However, if direct access storage is assigned to a system unit function, a matching entry for the function must be made in the 24-word Auxiliary Disk/Drum Limits Table that immediately follows the Auxiliary System Unit Function Table. A symbolic listing of the distributed version of this table is shown in Figure 39. In core storage, the Auxiliary Disk/Drum Limits Table begins at 15170_8 (SYSLB1) and ends at 15217_8 (SYSUT9). Note that in Figure 39 the locations corresponding to the SYSCRD (15174_8), SYSPRT (15175_8), and SYSPCH (15176_8) functions are used for purposes unrelated to the Disk/Drum Limits Table. Direct access storage would never be assigned to these functions.

An entry in the Auxiliary Disk/Drum Limits Table has the following format:

PZE dorg,,dend

where dorg and dend are the numbers of the first and last tracks, respectively, of the consecutive tracks in the direct access storage module (which is specified in the corresponding entry in the Auxiliary System Unit Function Table) that are to be assigned to the system unit function.

For example, to permanently assign tracks 0040 through 0079 of 1301 disk storage module CD01/0 (channel C, Access 0, Module 1, Data Channel Switch setting 1) as SYSUT3, the following octal alteration cards would be used in a job deck similar to the one shown in Figure 30:

15161 *OCT 023101000003
15211 *OCT 000117000050

To permanently assign tracks 0040 through 0079 of drum storage module CN02/0 (channel C, Access 0, Module 2, Data Channel Switch Setting 1) as SYSUT3, the first of the two octal alteration cards above would be changed to

15161 *OCT 023302000003

If the user wishes to maintain the Home Address 2 identifiers (HA2's) as they are defined in the HA2 Table (HA2TBL), no additional modifications are necessary. However, if the user wishes to change the HA2 definition for one or more system unit functions, the corresponding entry or entries in the HA2 Table must be modified. For example, to define the HA2 of SYSUT3 as PQ rather than 00 (*OCT 1212), the corresponding HA2 Table entry must be modified as follows:

xxxxx *OCT 004750000000

where xxxxx is the octal location of the SYSUT3 entry in the HA2 Table.

3	4	5—8	9—11	12	13	14—17
Reserve Flag (0)	Channel Type (1 for 7909)	Channel (1-8)	Device (000 for Hypertape) (001 for 1301 Disk) (011 for 1320 Drum) (111 for 2302 Disk)	Data Channel Switch (0 or 1)	Access (0 or 1)	Unit or Module Numbers (0-9 for Hypertape or Disk) (0, 2, 4, 6, or 8 for Drum)

Figure 31. Decrement of Entry for Hypertape, Disk, or Drum in Auxiliary System Unit Function Table

Preparing Duplicate 729 System Library Tapes for Alternate Use by the System Monitor

To reduce delays in processing due to rewinding of the System Library Tape, the complete 7090/7094 IBSYS Operating System can be duplicated on two 729 System Library Tapes that are used alternately by the System Monitor. A second, and alternate, method of reducing tape positioning time is described in the publication *IBM 7090/7094 IBSYS Operating System: IBJOB Processor*, Form C28-6389. With the alternate method, components of the IBJOB Processor are placed on a second System Library Tape which, in conjunction with a prepositioning feature in the IBJOB Processor, reduces delays in processing due to positioning of the System Library Tape. Heavy users of the IBJOB Processor should consider using the method described in the IBJOB Processor manual as an alternative to the method described in this manual.

When duplicate System Library Tapes are used, one tape is placed on the 729 Magnetic Tape Unit assigned as SYSLB1 and the other is placed on the unit assigned as SYSLB4. The entry in the Auxiliary System Unit Function Table for SYSLB4 on both tapes must contain the following:

```
MZE        channel,1,unit
```

where channel is the number of the channel (1 or 2 on the distributed preassembled tape) and unit is the number of the 729 Magnetic Tape Unit (1 through 10) on that channel to be assigned as SYSLB4. The 1 in the tag portion of the auxiliary entry indicates to the System Monitor that SYSLB4 does, in fact, contain a duplicate System Library Tape. The System Monitor will not alternate between duplicate System Library Tapes if the tag portion of the SYSLB4 entry is zero or if either SYSLB1 or SYSLB4 or both are not 729 Magnetic Tape Units.

```
1     7        16
-------------------------------------------------------
$JOB           DUPLICATE TAPES FOR PING PONG
$IBSYS
$IBEDT
      *EDIT
      *MODIFY  IBSYS
15143 *OCT     400005100001
   (END-OF-FILE CARD)
$IBSYS
$ATTACH        A3
$AS            SYSLB2
$ATTACH        XK
$AS            SYSUT1
$IBEDT
      *EDIT    SYSLB2
   (END-OF-FILE CARD)
$STOP

   NOTE, XK CAN BE ANY AVAILABLE 729 TAPE UNIT
```

Figure 32. Sample Job Deck for Preparing Duplicate System Library Tapes for Alternate Use by the System Monitor

The sample job deck in Figure 32 may be used to produce two duplicate System Library Tapes, in which the SYSLB4 auxiliary entry on both tapes is changed, as previously described, to allow alternate use of the tapes by the System Monitor. This deck will assign A5 as SYSLB4. It assumes that A3 is assigned initially as SYSUT1. The first of the two duplicate tapes will be produced on A3 (SYSUT1). A3 is then attached as SYSLB2 and is used to produce the second of the two duplicate tapes on a 729 Magnetic Tape Unit that is attached as SYSUT1. The 729 Magnetic Tape Unit specified on the second $ATTACH card can be any available unit. At the completion of the job, one of the two duplicate System Library Tapes will be on A3 and the other will be on the unit that is specified on the second $ATTACH card.

If the System Monitor is reassembled, as described in the section "Reassembling the System Monitor," alternate use of duplicate System Library Tapes can be specified by the assembly parameter PP SET cuu. This parameter is described in Figure 35, under the subheading "Miscellaneous Assembly Parameters," and is listed in Figure 36.

Adding 1301/2302 Disk Storage, 7320 Drum Storage, and/or 7340 Hypertape Capability

The distributed 729 capability System Library Tape, together with Symbolic Tape Number 1, may be used to produce a System Library with 1301/2302 Disk Storage, 7320 Drum Storage, and/or 7340 Hypertape capability, in addition to 729 capability, for the System Monitor, the IBJOB Processor, the Input/Output Control System, the Utilities, the Generalized Sorting System, and the Restart Program. The complete 7090/7094 Operating System can reside on direct access storage. Although each of the subsystems under IBSYS has this capability, the complete system cannot physically reside on one 7320 Drum Storage Unit. However, if a System Library is produced with Hypertape capability for the System Monitor, the IBJOB Processor, the Input/Output Control System, the Utilities, the Generalized Sorting System, and the Restart Program, these portions of the Operating System may reside on Hypertape, but the remainder of the Operating System, if it is to be used, must reside on 729 magnetic tape. To produce a System Library with disk, drum, and/or Hypertape capability, portions of the 7090/7094 IBSYS Operating System on the symbolic tape must be updated (to change assembly parameters), assembled, and then, together with the remainder of the 7090/7094 IBSYS Operating System, edited onto a new System Library Tape. These functions may be performed using a Disk/Drum/Hypertape Update-Edit Deck that is distributed with the 7090/7094 IBSYS Operating System. This deck is

```
$JOB    UPDATE EDIT DECK FOR 1301 DISK, 7320 DRUM AND 7340 HYPERTAPE
$IBSYS
$LIST
$ATTACH        A4
$AS            SYSCK1        UPDATE INTERMEDIATE TAPE
$REWIND        SYSUT4        UPDATE OUTPUT TAPE
$REWIND        SYSUT1        UPDATE INPUT TAPE
$REWIND        SYSCK1
$*
$*      NEW SYMBOLIC TAPE NUMBER 1 WITH DISK, DRUM OR
$*      HYPERTAPE PARAMETERS WILL BE WRITTEN ON TAPE UNIT B4.
$*
$EXECUTE       UPDATE
        UPDATE 4,3
SYSORG SET     2000                                               *ALL   1BB00250
                             SECTION 1 PARAMETERS
$STOP                                                                    1BD17710
        ENDFIL 3
        ENDUP
        UPDATE 4,3
CH7909 SET     1                                                  *ALL   1CJ01000
                             SECTION 2 PARAMETERS
        END                                                       *TRAIL 1I099990
        ENDUP                                                     *TRAIL
        UPDATE 4,9                                                *TRAIL
CH7909 SET     1                                                  *ALL   1PA01250
$STOP                                                             *TRAIL 1PC99999
        ENDFIL 9                                                  *TRAIL
        REWIND 9                                                  *TRAIL
        ENDUP                                                     *TRAIL
        UPDATE 9,3                                                *TRAIL
        END                                                       *TRAIL 1PC99990
        REWIND 9                                                  *TRAIL
        ENDUP                                                     *TRAIL
        UPDATE 9,3                                                *TRAIL
TRAIL  SET     1                                                  *TRAIL 1PA01374
       TTL     PROCESS CONTROL UTILITIES-IBJOBY                   *TRAIL 1PC12890
       BCI     1,IBJOBY                                           *TRAIL 1PC13790
       END                                                        *TRAIL 1PC99990
       REWIND  9                                                  *TRAIL
       ENDUP                                                      *TRAIL
       UPDATE  9,3                                                *TRAIL
TRAIL  SET     1                                                  *TRAIL 1PA01374
       TTL     PROCESS CONTROL UTILITIES-IBJOBI                   *TRAIL 1PC12890
       BCI     1,IBJOBI                                           *TRAIL 1PC13790
       END                                                        *TRAIL 1PC99990
       REWIND  9                                                  *TRAIL
       ENDUP                                                      *TRAIL
       UPDATE  9,3                                                *TRAIL
TRAIL  SET     1                                                  *TRAIL 1PA01374
       TTL     PROCESS CONTROL UTILITIES-IBJOBL                   *TRAIL 1PC12890
       BCI     1,IBJOBL                                           *TRAIL 1PC13790
$STOP                                                                    1PC99999
       ENDFIL  3
       REWIND  9                                                  *TRAIL
       ENDUP
       UPDATE  4,3
                             SECTION 3 PARAMETERS
       END                                                               1OR15470
       ENDFIL  3
       ENDUP
       UPDATE  4,3
$STOP                                                                    ED099999
       ENDFIL  3
       ENDUP
       UPDATE .4,3
                             SECTION 4 PARAMETERS
HYPER  SET     1                                                  *HYPR  3AK00210
DISK   SET     1                                                  *DKDM  3AK00230
HYPER  SET     1                                                  *HYPR  3AL00270
DISK   SET     1                                                  *DKDM  3AL00290
DISK   SET     1                                                  *DKDM  3B000370
       END                                                               3RN15370
       ENDFIL  3

       REWIND  3
       UNLOAD  4
       ENDUP
$IBSYS
$*
$*      PLEASE REMOVE THE DISTRIBUTED SYMBOLIC TAPE NO. 1 FROM TAPE UNIT
$*      A3 AND MOUNT A SCRATCH TAPE IN ITS PLACE.
$*
$*      A SYMBOLIC TAPE WITH ONLY THOSE RECORDS WHICH MUST BE REASSEMBLED
$*      FOR DISK, DRUM OR HYPERTAPE WILL NOW BE WRITTEN ON TAPE UNIT A3.
$*      PLEASE PRESS START WHEN ALL TAPE MOUNTING IS COMPLETE.
$*
$PAUSE
$EXECUTE       UPDATE
       UPDATE  3,4                                                       1BD17690
```

Figure 33. Distributed Disk/Drum/Hypertape Update-Edit Deck (Sheet 1 of 2)

```
        END     *                                                         IBD17690
        ENDUP
        UPDATE  3,4
        SKIPTO                                                            1CJ00010
        END                                                               1CT99990
        SKIPTO                                                            11000000
        END                                                               1PC99990
        ENDUP
        UPDATE  3,4                                               *TRAIL
        END                                                       *TRAIL  1PC99990
        ENDUP                                                     *TRAIL
        UPDATE  3,4                                               *TRAIL
        END                                                       *TRAIL  1PC99990
        ENDUP                                                     *TRAIL
        UPDATE  3,4                                               *TRAIL
        END                                                       *TRAIL  1PC99990
        ENDUP                                                     *TRAIL
        UPDATE  3,4
        SKIPTO                                                            IOA00000
$IBSYS
$JOB            7090-IO-919, INPUT OUTPUT CONTROL SYSTEM, VERSION 6
$EXECUTE        IBSFAP
*       FAP
        END     -1                                                        IOF00510
        ENDUP
        UPDATE  3,4
        SKIPTO                                                            3AC00000
$IBSYS
$JOB            IBJOB SUBROUTINE LIBRARY, 7090-LM-803, VERSION 5
$EXECUTE        IBJOB
$IBJOB IBLIB    MAP,LOGIC
$EDIT           LOGIC
$REPLACE        .JBCON,ORG=03720
$IEDIT
        END                                                               3AC00200
        SKIPTO                                                            3AE00000
$REPLACE        .IODEF
        END                                                               3AE01990
        SKIPTO                                                            3AF00000
$REPLACE        .IOCSF
        END                                                               3AF28140
        SKIPTO                                                            3AK00000
$REPLACE        .LOVRY
        END     .LOVRY    IF THIS DECK IS TO BE RUN USING SYMBOLIC  ----> 3AK03400
        SKIPTO            TAPE NUMBER 1 CONTAINING VERSION 5 OF           3AL00000
$REPLACE        .LXSL     IBLIB AT A MODIFICATION LEVEL GREATER
        END               THAN 3,THE SERIALIZATION MUST BE                3AL02520
        SKIPTO            3AK10100.                                       3B000000
$REPLACE        .IBDBI
        END     DSTRS                                                     3B020280
        SKIPTO                                                            31000000
$REPLACE        .IOCS
        END                                                               31056800
        ENDFIL  4
        UNLOAD  3
        ENDUP
        UPDATE  ,4,U
$IBSYS
$SWITCH         SYSPP1,SYSUT2
$IBEDT
       *EDIT     MAP,MODS
TAPE   *REPLACE IBSYS
TAPE   *REPLACE SYSDMP
TAPE   *REPLACE IBJOB
TAPE   *REPLACE IOCSB
TAPE   *REPLACE IBJOBB
TAPE   *REPLACE IBJOBC
       *AFTER    IBCBC9                                           *TRAIL
TAPE   *INSERT   IBJOBY                                           *TRAIL
       *AFTER    IBFTCG
TAPE   *INSERT   IBJOBI                                           *TRAIL
       *AFTER    IBMAPK                                           *TRAIL
TAPE   *INSERT   IBJOBL                                           *TRAIL
FILE   *REPLACE CIFSR,SYSUT4
FILE   *REPLACE TIFSR,SYSUT4
TAPE   *REPLACE IOCS
TAPE   *REPLACE POST
TAPE   *REPLACE PREP
TAPE   *REPLACE IOBB
TAPE   *REPLACE IOBM
TAPE   *REPLACE NOBS
        ENDFIL  4
$IBSYS
$*
$*      TAPE UNIT A3 NOW CONTAINS THE NEW SYSTAP
$*      WITH DISK, DRUM OR HYPERTAPE CAPABILITIES.
$*
$STOP
        REWIND  4
        ENDUP
$IBSYS
$*
$*      REWIND AND SWITCH TAPE UNITS A2 AND A3.
$*      DISMOUNT AND SAVE NEW SYMBOLIC TAPE ON B4 AND MOUNT A SCRATCH.
$*      REWIND SYSTAP, CLEAR AND LOAD TAPE TO ASSEMBLE AND EDIT A SYSTAP
$*      WITH DISK, DRUM OR HYPERTAPE CAPABILITY.
$*
$*
$*
$REWIND          SYSPP1
$PAUSE
```

Figure 33. Distributed Disk/Drum/Hypertape Update-Edit Deck (Sheet 2 of 2)

listed in Figure 33. The portions of the 7090/7094 IBSYS Operating System that are updated by the deck are contained on the 7090/7094 IBSYS Operating System Symbolic Tape Number 1. The contents of this tape are as follows:

File 1	System Monitor including the System Core-Storage Dump Program
File 2	IBJOB Monitor
File 3	Full IOCS (Independent IOCS)
File 4	System Editor
File 5	IBJOB Subroutine Library (IBLIB)

Operations Performed by the Disk/Drum/Hypertape Update-Edit Job Deck

Before using the Disk/Drum/Hypertape Update-Edit Deck, the user must prepare the deck (as described later in the text) to ensure a logical configuration of input/output units tailored to the physical limitations and operational requirements of his installation. The following sequence of operations is performed by the update-edit job deck:

1. The portions of the 7090/7094 IBSYS Operating System on Symbolic Tape Number 1 that contain disk, drum, and/or Hypertape assembly parameters are updated with the appropriate changes by the Symbolic Update Program. Other portions of Symbolic Tape Number 1 that do not contain these parameters are copied without change onto the update output tape. The IBJOB trailer records IBJOBY, IBJOBI, and IBJOBL are also copied onto the update output tape by means of special updating of the IBJOB record on Symbolic Tape Number 1.

This special form of Symbolic Tape Number 1 contains five files, updated where necessary for disk, drum, and/or Hypertape capability. These five files are:

File 1	System Monitor including the System Core-Storage Dump Program
File 2	IBJOB Monitor and the IBJOBY, IBJOBI, and IBJOBL records
File 3	Full IOCS
File 4	System Editor
File 5	IBJOB Subroutine Library (IBLIB)

2. A second and final update is performed which extracts those portions of Symbolic Tape Number 1 which must actually be assembled for disk, drum, and/or Hypertape capability. Systems or portions of systems which are not needed for reassembly are deleted from this final symbolic tape, and the appropriate control cards are added to allow this tape to be used as a system input tape, which will assemble and edit the appropriate systems onto a new System Library Tape having disk, drum, and/or Hypertape capability. The updated Symbolic Tape Number 1 created during the job can now be removed and retained for use with any future IBM modifications to the subsystems and components contained on this tape.

3. The selectively constructed system input tape on tape unit A3 is now interchanged manually with tape unit A2 (the original system input tape). This tape now becomes SYSIN for the remainder of the job.

4. The IBSYS Monitor and the IBJOB Monitor, together with the IBJOB trailer records, if necessary, are assembled by the IBJOB Processor. Independent IOCS is assembled by the FORTRAN II Assembly Program (IBSFAP). The IBJOB Processor then assembles and edits the portions of the IBJOB Subroutine Library containing the updated disk/drum/Hypertape assembly parameters.

5. The System Editor produces a new System Library Tape on tape unit A3 with the required disk, drum, and/or Hypertape capability by replacing the IBSYS Monitor, the IBJOB Monitor, the IBJOB Subroutine Library, and Independent IOCS from the distributed System Library Tape with their reassembled disk/drum/Hypertape counterparts.

The Assignment and Function of Units for the Disk/Drum/Hypertape Update-Edit Deck

As distributed, the Disk/Drum/Hypertape Update-Edit Deck requires eight 729 Magnetic Tape Units. If a user has less than eight 729 Magnetic Tape Units, he should request his IBM representative to obtain a special procedure from the regional Programming Systems Support Group.

The Disk/Drum/Hypertape Update-Edit Deck is designed for use with the distributed System Library Tape with the input/output units initially assigned to system unit functions, as shown in Figure 28. The assignment and function of the units for the Disk/Drum/Hypertape Update-Edit Deck are as shown in Figure 34.

Unit	Initial Assignment	Additional Assignment	Function	Logical Number
A1	SYSLB1	None	Old System Library	1
A2	SYSIN1	None	System Input	5
A3	SYSUT1	None	Update Input Intermediate New System Library	4
A4	SYSUT3	SYSCK1	Intermediate	2, 9
B1	SYSOU1	None	List Output	6
B2	SYSPP1	SYSUT2	Punch Output	7
B3	SYSUT2	SYSPP1	Intermediate	8
B4	SYSUT4	None	Update Output Intermediate	3

Figure 34. The Assignment and Function of Units for the Disk/Drum/Hypertape Update-Edit Deck

The distributed System Library Tape is mounted on A1, the Symbolic Tape Number 1 is mounted on A3, and the system input tape (containing the update-edit deck) is mounted on A2.

The tape written on B4 and the final tape written on A3 should be saved. These two tapes contain the following:

Tape on B4: Updated Symbolic Tape Number 1 containing the System Monitor, the IBJOB Monitor, and possibly the IBJOBY, IBJOBI, and IBJOBL trailer records (see the section "Removal and Replacement of Cards in the Distributed Deck"), Full IOCS (Independent IOCS), the System Editor, and the IBJOB Subroutine Library (IBLIB).

Tape on A3: New System Library Tape in binary form, containing the complete 7090/7094 IBSYS Operating System with disk, drum, and/or Hypertape capability for the System Monitor, the IBJOB Processor, the Input/Output Control System, the Utilities, and the Restart Program.

Preparing the Disk/Drum/Hypertape Update-Edit Deck

Before using the distributed update-edit deck, certain control cards and assembly parameter cards already in the deck may have to be removed or replaced, and assembly parameter cards and cards for changing system unit assignment must be inserted in the deck. The insertion of these cards establishes the logical configuration of disk, drum, and/or Hypertape units required for a specific installation. Although the distributed job deck is designed primarily for changing the operating characteristics of the system to achieve direct access storage and/or Hypertape capability, System Monitor assembly parameters can, if necessary, be inserted in the deck in order to change (during the same job run) operating characteristics of the system that are not directly related to disk, drum, or Hypertape.

Assembly parameters that may be inserted in the update-edit job deck, together with the entries in the Auxiliary System Unit Function Table and the Auxiliary Disk/Drum Limits Table, are listed in Figures 36 through 43 as they appear in the distributed IBSYS Symbolic Tape Number 1. They are listed in the relative order in which they would be placed in the job deck. Assembly parameters that are related to disk, drum, or Hypertape capability contain *DISK, *DRUM, or *HYPR, respectively, in columns 67 through 71. Parameters that are related to both disk and drum contain *DKDM in columns 67 through 71. Assembly parameters that are not directly related to either disk, drum, or Hypertape capability contain blanks in columns 67 through 71. The use of these parameters is optional. Other assembly parameters, which merely indicate whether or not disk, drum, and/or Hypertape capability are desired, are already included in the distributed update-edit deck. Most of these parameters are required for disk, drum, and Hypertape capability and therefore contain *ALL in columns 67 through 70.

Assembly parameter cards listed in Figures 36 through 43 are grouped into four sections for insertion in the update-edit deck. These sections are numbered from 1 through 4 and the section to which each type of assembly parameter card belongs is indicated in the title of the figure in which it is shown. The places at which assembly parameters of each section are to be inserted in the update-edit deck are indicated by four cards containing the following message:

SECTION x PARAMETERS

where x is either 1, 2, 3, or 4. As assembly parameter cards are inserted in the deck, these four cards should be removed. However, before inserting assembly parameter cards, certain cards already in the deck may be removed or replaced, depending on the requirements of an installation. The following procedure should be followed to prepare the distributed update-edit deck for use.

Removal and Replacement of Cards in the Distributed Deck

The IBSYS Symbolic Tape Number 1 is distributed in either of three densities: 800 cpi, 556 cpi, or 200 cpi. The user must verify that the Disk/Drum/Hypertape Update-Edit Deck will read SYSUT1 (the distributed Symbolic Tape Number 1) in the correct density. In the distributed deck, high density is assumed for all units. If the user requires the System Library Tape, created as a result of running the update-edit job deck, to be at 800 cpi, Symbolic Tape Number 1 should be ordered at 800 cpi. If, for some reason, the user finds working with 800 cpi tapes undesirable, Symbolic Tape Number 1 may be ordered at 556 cpi, and the update-edit deck may be run with the computer in Mode B (556/200 cpi). The result of operating in this manner will be the creation of a new System Library Tape written at 556 cpi.

If the System Library Tape generated by the update-edit deck is to reside on direct access storage exclusively, the direct access storage requirements may be reduced by removing the cards in the distributed deck which cause the IBJOBY, IBJOBI, and IBJOBL trailer records of the IBJOB Monitor to be assembled and edited onto the new System Library Tape. To accomplish this, remove all cards containing *TRAIL in card columns 66-71. If this change is made in the distributed deck, the updated Symbolic Tape Number 1 will not contain the IBJOBY, IBJOBI, and IBJOBL records. If the System Library produced by the update-edit deck is ever to reside on a 729 Magnetic Tape Unit or 7340 Hypertape Drive, the changes described above should not be made.

Immediately following the insertion point in the update-edit deck for the SECTION 4 PARAMETERS are five parameter cards. If only direct access storage capabilities are to be added, the two cards containing *HYPR in columns 67 through 71 should be removed from the deck. If only 7340 capability is to be added, the three

Symbol	Code	Variable	Description
			Miscellaneous Assembly Parameters
IBM	SET	x	The decimal number x specifies the machine type; 709 for IBM 709 and 7090 for IBM 7090/7094. If IBM SET 709, the instructions that refer to the "Set Density" operations in (NDATA are deleted).
IBSORG	SET	x	The decimal number x is the origin of the System Nucleus. It cannot be less than 64.
IOXORG	SET	x	The decimal number x is the origin of IOEX. It cannot be less than the location of the last word of the last unit control block plus one.
SYSORG	SET	x	The decimal number x is the common subsystem origin. It cannot be less than the last location used by IOEX plus 51. For disk, drum and /or Hypertape capability SYSORG should be at least 2000.
SYSEND	SET	x	The decimal number x is the common upper core limits for all subsystems. SYSEND+1 should be the origin of an installation accounting routine if it exists.
HIGHLO	SET	x	The decimal number x specifies the density to be assigned to a unit if the density specification is omitted from an $AS card. If x is 1, high density is assigned. If x is 0, low density is assigned.
PP	SET	cuu	This parameter may be used to specify alternate use of duplicate 729 System Library tapes on SYSLB1 and SYSLB4. The decimal digit c is the channel number (1 through 8) and uu is the number of the 729 tape unit (1 through 10) on that channel to be assigned as SYSLB4. If PP SET 0 is specified, the System Monitor will not alternate between duplicate System Library Tapes.
EJECT	SET	x	The decimal number x specifies the printer board sense exit (1 through 10) for an eject.
DBLSP	SET	x	The decimal number x specifies the printer board sense exit (1 through 10) for a double space.
RDUNRT	SET	x	The decimal number x is the number of recovery attempts IOEX should make on a read redundancy before the redundancy is considered permanent.
ETMODE	SET	x	If the decimal number x is 1, trap mode will be saved by IOEX. If x is 0, trap mode will remain in effect in IOEX.
			Input/Output Configuration Assembly Parameters
CHA1	SET	x	The decimal number x is the number of 729 Magnetic Tape Units existing on channel A.
CHAAT	BOOL	xxxxx	The rightmost 13 bits of xxxxx correspond to the card reader, card punch, printer, and 729 Magnetic Tape Units 1 through 10, respectively, beginning with the leftmost of the 13 bits. If a bit is a zero, the corresponding unit is attached to the channel; a one-bit specifies a detached unit.
CHAMD	BOOL	xxxx	The rightmost 10 bits of xxxx correspond to 729 Magnetic Tape Units 1 through 10, respectively, beginning with the leftmost of the 10 bits. If a bit is zero, the model type of the corresponding 729 Magnetic Tape Unit is specified as II or V; a one-bit specifies a IV or VI model type.
PRNTA	SET	x	If the decimal number x is a 1, the printer is considered to exist on channel A. If no printer exists on channel A, x should be 0.
PNCHA	SET	x	If a card punch exists on channel A, the decimal number should be 1; otherwise x should be 0.
CDRDA	SET	x	If a card reader exists on channel A, the decimal number should be 1; otherwise x should be 0. NOTE: A unit control block is generated at initial start for every unit that exists on channel A as specified by CHA1 for 729 Magnetic Tape Units, PRNTA for the printer, PNCHA for the card punch, and CDRDA for the card reader. The parameter CHAAT merely specifies that the unit control block that is generated should indicate that the unit is either attached or detached.
HTA1	SET	x	The decimal number x specifies the number of 7340 Hypertape Drives (0 through 10) that exist on Data Channel Switch setting 1 for channel A.

Symbol	Code	Variable	Description
			Input/Output Configuration Assembly Parameters
HTAA1	BOOL	xxxx	The rightmost 10 bits of xxxx specify which of the Hypertape drives (specified in HTA1) on Data Channel Switch setting 1 are attached, beginning with the leftmost of the ten bits. If a bit is a zero, the corresponding drive is attached to the channel; a one-bit specifies a detached unit. For example, if there are eight Hypertape drives on channel A, Data Channel Switch setting 1, and only six of the eight are to be attached, the following should be specified: HTA1 SET 8 HTAA1 BOOL 0017 Although the two units in this example, AH6/0 and AH7/0, will not be attached at initial start, they may be temporarily attached by means of a $ATTACH control card. Units AH8/0 and AH9/0, however, cannot be attached with a $ATTACH card since HTA1 SET 8 and, therefore, no unit control block will be generated for them at initial start.
HTA2	SET	x	HTA2 is the same as HTA1 except that it applies to Hypertape drives on Data Channel Switch setting 2.
HTAA2	BOOL	xxxx	HTAA2 is the same as HTAA1 except that it applies to Hypertape drives (specified in HTA2) on Data Channel Switch setting 2.
DFA1	SET	x	The decimal number x specifies the number of 1301 Disk Storage modules (0 through 10) that exist on Data Channel Switch setting 1 for channel A.
DFA3	SET	x	The decimal number x specifies the number of 2302 Disk Storage modules (0 through 10) that exist on Data Channel Switch setting 1 for channel A.
NFA1	SET	x	The decimal number x specifies the number of 7320 Drum Storage modules (0 through 5) that exist on Data Channel Switch setting 1 for channel A.
DFAA1	BOOL	xxxx	The rightmost 10 bits of xxxx specify which of the disk or drum modules (specified in DFA1 and NFA1) on Data Channel Switch Setting 1 are attached, beginning with the leftmost of the 10 bits. If a bit is zero, the corresponding module is attached; a one-bit specifies a detached module.
IFA1	SET	x	The decimal number x specifies the 7631 model type for disk and drum modules on Data Channel Switch setting 1. If the model type is III or IV, x is 1; otherwise x is 0.
DFA2	SET	x	DFA2 is the same as DFA1 except that is applies to disk modules on Data Channel Switch setting 2.
DFA4	SET		DFA4 is the same as DFA3 except that it applies to disk modules (specified in DFA2) on Data Channel Switch setting 2.
NFA2	SET	x	NFA2 is the same as NFA1 except that it applies to drum modules on Data Channel Switch setting 2.
DFAA2	BOOL	xxxx	DFAA2 is the same as DFAA1 except that it applies to disk and drum modules (specified in DFA2 and NFA2) on Data Channel Switch setting 2.
IFA2	SET	x	IFA2 is the same as IFA1 except that it applies to disk and drum modules on Data Channel Switch setting 2.
			System Core-Storage Dump Program Assembly Parameters
ONLIN	BOOL	x	This parameter determines whether dump output will be listed on the System Output Unit only or whether the tag portion of the control word (Figure 5) or console entry keys will determine the output unit. If x is 0, the tag will determine the output unit.
DBLSPC	BOOL	x	If x is 0, dump output will be single-spaced. If x is 1, output will be double spaced.
KEYSWT	SET	x	The decimal number x determines the sense switch (1-6) that will be tested for the console entry keys option.
FORMAT	SET	x	The decimal number x determines the output format (1-6) that will be used by the dump program when none is specified. Output formats and their numbers are shown in Figure 6.

Figure 35. Explanation of System Monitor Assembly Parameters

IBM	SET	7090	IBB00130
IBSORG	SET	64	IBB00170
IOXORG	SET	450	IBB00210
SYSORG	SET	1450	IBB00250
SYSEND	SET	-1	IBB00290
HIGHLO	SET	1	IBB00330
PP	SET	0	IBB00370
EJECT	SET	1	IBB00420
DBLSP	SET	4	IBB00450
RDUNRT	SET	100	IBB00480
ETMODE	SET	0	IBB00530

Figure 36. Miscellaneous System Monitor Assembly Parameters (SECTION 1)

cards containing *DKDM in columns 67 through 71 should be removed. If disk, drum, and Hypertape capabilities are all to be added, none of the five cards should be removed from the deck.

Inserting Miscellaneous System Monitor Assembly Parameters

If any changes are to be made to the distributed version of the miscellaneous System Monitor assembly parameters in Figure 36, the appropriate cards should be punched and inserted as part of the SECTION 1 PARAMETER cards in the update-edit deck. These parameters are described in Figure 35. For disk, drum, and/or Hypertape capability, the SYSORG parameter must be at least 2000_{10}. Therefore, a card with SYSORG equal to 2000_{10} is included in the distributed update-edit deck. If other miscellaneous System Monitor assembly parameter cards are to be inserted in the deck, they should precede or follow the SYSORG parameter card, depending on their serialization.

Inserting Input/Output Configuration Assembly Parameters

The input/output configuration assembly parameters (shown in Figure 37) that are to be changed are inserted as part of the SECTION 1 PARAMETER cards in the update-edit deck, immediately following the miscellaneous System Monitor assembly parameters. These parameters are described in detail in Figure 35. Since the parameters for each channel are similar, only the parameters for channel A are described in Figure 36. The parameters that are changed should be inserted in the deck in the same relative order as they are listed in Figure 37.

Inserting Changes to Entries in the Auxiliary System Unit Function and Disk/Drum Limits Tables

Changes to entries in the Auxiliary System Unit Function Table (Figure 38) are inserted as part of the SECTION 1 PARAMETER cards in the update-edit deck, immediately following the input/output configuration assembly parameters. These changes are followed by any changes to the Auxiliary Disk/Drum Limits Table (Figure 39).

Changing the Assignment of 729 Magnetic Tape Units and Card Equipment: Each entry for 729 Magnetic Tape Units and card equipment in the Auxiliary System Unit Function Table has the following format:

pfx   channel,,unit

where pfx is the density assigned to the system unit function: MZE for high density and PZE for low density; and channel is a number from 1 through 8 (corresponding to channels A through H) which indicates the channel of the unit assigned to the system unit function. The value unit is the number of the 729 Magnetic Tape Unit (1 through 10), card reader (11), card punch (12), or printer (13) assigned to the system unit function. As an example, the following card would be inserted in the update-edit deck to assign 729 Magnetic Tape Unit B5 as SYSOU2 at low density:

```
SYSOU2     PZE     2,,5                IBB42360
```

Changing the Assignment of 7340 Hypertape Drives: Each entry in the Auxiliary System Unit Function Table has the following format:

PZE   channel,,unit

where channel is a number from 1 through 8, corresponding to channels A through H, and unit is the full address of the Hypertape drive specified in the decrement portion of the first word of its unit control block, as shown in Figure 31. The decimal equivalent of a Hypertape drive address may be specified in the decrement portion of an entry, or a BOOL pseudo-operation may be used to define the address. For example, to assign HH9/1 (channel H, Hypertape drive 9, Data Channel Switch setting 2) as SYSUT2, the following cards would be inserted among the SECTION 1 PARAMETER cards in the update-edit deck:

```
H          BOOL    30051               IBB42439
SYSUT2     PZE     8,,H                IBB42440
```

Changing the Assignment of 1301/2302 Disk Storage and 7320 Drum Storage: An entry in the Auxiliary System Unit Function Table for disk or drum storage is similar to an entry for Hypertape except that bits 14 through 17 specify a module number (0-9 for disk; 0, 2, 4, 6, or 8 for drum), rather than a unit number, and bits 9 through 11 are 001 for 1301 disk, 111 for 2302 disk, and 011 for drum, instead of 000 (see Figure 31). For example, to assign 1301 disk storage module

```
                                   CHANNEL A
CHA1     SET     10                                                      IBB00580
CHAAT    BOOL    00003                                                   IBB00630
CHAMD    BOOL    01777                                                   IBB00670
PRNTA    SET     1                                                       IBB00700
PNCHA    SET     1                                                       IBB00730
CDRDA    SET     1                                                       IBB00760
HTA1     SET     0                                                  *HYPR IBB00800
HTAA1    BOOL    1777                                               *HYPR IBB00820
HTA2     SET     0                                                  *HYPR IBB00850
HTAA2    BOOL    1777                                               *HYPR IBB00870
DFA1     SET     0                                                  *DISK IBB00910
DFA3     SET     1                                                  *DISK IBB00930
NFA1     SET     0                                                  *DRUM IBB00950
DFAA1    BOOL    1777                                               *DKDM IBB00970
IFA1     SET     0                                                  *DKDM IBB00990
DFA2     SET     0                                                  *DISK IBB01020
DFA4     SET     1                                                  *DISK IBB01040
NFA2     SET     0                                                  *DRUM IBB01060
DFAA2    BOOL    1777                                               *DKDM IBB01080
IFA2     SET     0                                                  *DKDM IBB01100
                                   CHANNEL B
CHB1     SET     10                                                      IBB01150
CHBAT    BOOL    16003                                                   IBB01200
CHBMD    BOOL    01777                                                   IBB01240
PRNTB    SET     0                                                       IBB01270
PNCHB    SET     0                                                       IBB01300
CDRDB    SET     0                                                       IBB01330
HTB1     SET     0                                                  *HYPR IBB01370
HTBA1    BOOL    1777                                               *HYPR IBB01390
HTBA2    BOOL    1777                                               *HYPR IBB01440
HTB2     SET     0                                                  *HYPR IBB01420
DFB1     SET     0                                                  *DISK IBB01480
DFB3     SET     1                                                  *DISK IBB01500
NFB1     SET     0                                                  *DRUM IBB01520
DFBA1    BOOL    1777                                               *DKDM IBB01540
IFB1     SET     0                                                  *DKDM IBB01560
DFB2     SET     0                                                  *DISK IBB01590
DFB4     SET     1                                                  *DISK IBB01610
NFB2     SET     0                                                  *DRUM IBB01630
DFBA2    BOOL    1777                                               *DKDM IBB01650
IFB2     SET     0                                                  *DKDM IBB01670
                                   CHANNEL C
CHC1     SET     6                                                       IBB01720
CHCAT    BOOL    17777                                                   IBB01770
CHCMD    BOOL    1777                                                    IBB01810
PRNTC    SET     0                                                       IBB01840
PNCHC    SET     0                                                       IBB01870
CDRDC    SET     0                                                       IBB01900
HTC1     SET     0                                                  *HYPR IBB01940
HTCA1    BOOL    1777                                               *HYPR IBB01960
HTC2     SET     0                                                  *HYPR IBB01990
HTCA2    BOOL    1777                                               *HYPR IBB02010
DFC1     SET     0                                                  *DISK IBB02050
DFC3     SET     1                                                  *DISK IBB02070
NFC1     SET     0                                                  *DRUM IBB02090
DFCA1    BOOL    1777                                               *DKDM IBB02110
IFC1     SET     0                                                  *DKDM IBB02130
DFC2     SET     0                                                  *DISK IBB02160
DFC4     SET     1                                                  *DISK IBB02180
NFC2     SET     0                                                  *DRUM IBB02200
DFCA2    BOOL    1777                                               *DKDM IBB02220
IFC2     SET     0                                                  *DKDM IBB02240
                                   CHANNEL D
CHD1     SET     6                                                       IBB02290
CHDAT    BOOL    17777                                                   IBB02340
CHDMD    BOOL    1777                                                    IBB02380
PRNTD    SET     0                                                       IBB02410
PNCHD    SET     0                                                       IBB02440
CDRDD    SET     0                                                       IBB02470
HTD1     SET     0                                                  *HYPR IBB02510
HTDA1    BOOL    1777                                               *HYPR IBB02530
HTD2     SET     0                                                  *HYPR IBB02560
HTDA2    BOOL    1777                                               *HYPR IBB02580
DFD1     SET     0                                                  *DISK IBB02620
DFD3     SET     1                                                  *DISK IBB02640
NFD1     SET     0                                                  *DRUM IBB02660
DFDA1    BOOL    1777                                               *DKDM IBB02680
IFD1     SET     0                                                  *DKDM IBB02700
DFD2     SET     0                                                  *DISK IBB02730
DFD4     SET     1                                                  *DISK IBB02750
NFD2     SET     0                                                  *DRUM IBB02770
DFDA2    BOOL    1777                                               *DKDM IBB02790
IFD2     SET     0                                                  *DKDM IBB02810
```

Figure 37. Input/Output Configuration System Monitor Assembly Parameters (SECTION 1)
(Sheet 1 of 2)

```
                                   CHANNEL E
CHE1    SET     0                                                              IBB02860
CHEAT   BOOL    17777                                                          IBB02910
CHEMD   BOOL    1777                                                           IBB02950
PRNTE   SET     0                                                              IBB02980
HTEA1   BOOL    1777                                                     *HYPR IBB03100
PNCHE   SET     0                                                              IBB03010
CDRDE   SET     0                                                              IBB03040
HTE1    SET     0                                                        *HYPR IBB03080
HTE2    SET     0                                                        *HYPR IBB03130
HTEA2   BOOL    1777                                                     *HYPR IBB03150
DFE1    SET     0                                                        *DISK IBB03190
DFE3    SET     1                                                        *DISK IBB03210
NFE1    SET     0                                                        *DRUM IBB03230
DFEA1   BOOL    1777                                                     *DKDM IBB03250
IFE1    SET     0                                                        *DKDM IBB03270
DFE2    SET     0                                                        *DISK IBB03300
DFE4    SET     1                                                        *DISK IBB03320
NFE2    SET     0                                                        *DRUM IBB03340
DFEA2   BOOL    1777                                                     *DKDM IBB03360
IFE2    SET     0                                                        *DKDM IBB03380
                                   CHANNEL F
CHF1    SET     0                                                              IBB03430
CHFAT   BOOL    17777                                                          IBB03480
CHFMD   BOOL    1777                                                           IBB03520
PRNTF   SET     0                                                              IBB03550
PNCHF   SET     0                                                              IBB03580
CDRDF   SET     0                                                              IBB03610
HTF1    SET     0                                                        *HYPR IBB03650
HTFA1   BOOL    1777                                                     *HYPR IBB03670
HTF2    SET     0                                                        *HYPR IBB03700
HTFA2   BOOL    1777                                                     *HYPR IBB03720
DFF1    SET     0                                                        *DISK IBB03760
DFF3    SET     1                                                        *DISK IBB03780
NFF1    SET     0                                                        *DRUM IBB03800
DFFA1   BOOL    1777                                                     *DKDM IBB03820
IFF1    SET     0                                                        *DKDM IBB03840
DFF2    SET     0                                                        *DISK IBB03870
DFF4    SET     1                                                        *DISK IBB03890
NFF2    SET     0                                                        *DRUM IBB03910
DFFA2   BOOL    1777                                                     *DKDM IBB03930
IFF2    SET     0                                                        *DKDM IBB03950
                                   CHANNEL G
CHG1    SET     0                                                              IBB04000
CHGAT   BOOL    17777                                                          IBB04050
CHGMD   BOOL    1777                                                           IBB04090
PRNTG   SET     0                                                              IBB04120
PNCHG   SET     0                                                              IBB04150
CDRDG   SET     0                                                              IBB04180
HTG1    SET     0                                                        *HYPR IBB04220
HTGA1   BOOL    1777                                                     *HYPR IBB04240
HTG2    SET     0                                                        *HYPR IBB04270
HTGA2   BOOL    1777                                                     *HYPR IBB04290
DFG1    SET     0                                                        *DISK IBB04330
DFG3    SET     1                                                        *DISK IBB04350
NFG1    SET     0                                                        *DRUM IBB04370
DFGA1   BOOL    1777                                                     *DKDM IBB04390
IFG1    SET     0                                                        *DKDM IBB04410
DFG2    SET     0                                                        *DISK IBB04440
DFG4    SET     1                                                        *DISK IBB04460
NFG2    SET     0                                                        *DRUM IBB04480
DFGA2   BOOL    1777                                                     *DKDM IBB04500
IFG2    SET     0                                                        *DKDM IBB04520
                                   CHANNEL H
CHH1    SET     0                                                              IBB04570
CHHAT   BOOL    17777                                                          IBB04620
CHHMD   BOOL    1777                                                           IBB04660
PRNTH   SET     0                                                              IBB04690
PNCHH   SET     0                                                              IBB04720
CDRDH   SET     0                                                              IBB04750
HTH1    SET     0                                                        *HYPR IBB04790
HTHA1   BOOL    1777                                                     *HYPR IBB04810
HTH2    SET     0                                                        *HYPR IBB04840
HTHA2   BOOL    1777                                                     *HYPR IBB04860
DFH1    SET     0                                                        *DISK IBB04900
DFH3    SET     1                                                        *DISK IBB04920
NFH1    SET     0                                                        *DRUM IBB04940
DFHA1   BOOL    1777                                                     *DKDM IBB04960
IFH1    SET     0                                                        *DKDM IBB04980
DFH2    SET     0                                                        *DISK IBB05010
DFH4    SET     1                                                        *DISK IBB05030
NFH2    SET     0                                                        *DRUM IBB05050
DFHA2   BOOL    1777                                                     *DKDM IBB05070
IFH2    SET     0                                                        *DKDM IBB05090
```

Figure 37. Input/Output Configuration System Monitor Assembly Parameters (SECTION 1) (Sheet 2 of 2)

```
SYSLB1 MZE     1,,1                   HIGH DENSITY A1                        IBB42280
SYSLB2 MZE     0                      NO UNIT ASSIGNED                       IBB42290
SYSLB3 MZE     0                      NO UNIT ASSIGNED                       IBB42300
SYSLB4 MZE     PPC,PPE,PPU            NO UNIT ASSIGNED                       IBB42310
SYSCRD PZE     1,,11                  CARD READER ON A                       IBB42320
SYSPRT PZE     1,,13                  PRINTER ON A                           IBB42330
SYSPCH PZE     1,,12                  PUNCH ON A                             IBB42340
SYSOU1 MZE     2,,1                   HIGH DENSITY B1                        IBB42350
SYSOU2 MZE     2,,1                   HIGH DENSITY B1                        IBB42360
SYSIN1 MZE     1,,2                   HIGH DENSITY A2                        IBB42370
SYSIN2 MZE     1,,2                   HIGH DENSITY A2                        IBB42380
SYSPP1 MZE     2,,2                   HIGH DENSITY B2                        IBB42390
SYSPP2 MZE     2,,4                   HIGH DENSITY B4                        IBB42400
SYSCK1 MZE     0                      NO UNIT ASSIGNED                       IBB42410
SYSCK2 MZE     1,,5                   HIGH DENSITY A5                        IBB42420
SYSUT1 MZE     1,,3                   HIGH DENSITY A3                        IBB42430
SYSUT2 MZE     2,,3                   HIGH DENSITY B3                        IBB42440
SYSUT3 MZE     1,,4                   HIGH DENSITY A4                        IBB42450
SYSUT4 MZE     2,,4                   HIGH DENSITY B4                        IBB42460
SYSUT5 MZE     0                      NO UNIT ASSIGNED                       IBB42461
SYSUT6 MZE     0                      NO UNIT ASSIGNED                       IBB42462
SYSUT7 MZE     0                      NO UNIT ASSIGNED                       IBB42463
SYSUT8 MZE     0                      NO UNIT ASSIGNED                       IBB42464
SYSUT9 MZE     0                      NO UNIT ASSIGNED                       IBB42465
```

Figure 38. Auxiliary System Unit Function Table (SECTION 1)

```
       PZE                                                                   IBB42510
       PZE                                                                   IBB42520
       PZE                                                                   IBB42530
       PZE                                                                   IBB42540
DMPSW  PZE     **                     DUMP SWITCH                            IBB42550
DMPSV  PZE     **,7,**                DUMP INFO SAVE                         IBB42560
IBSAV  PZE     **                     S.S.1 SAVE                             IBB42570
       PZE                                                                   IBB42580
       PZE                                                                   IBB42590
       PZE                                                                   IBB42600
       PZE                                                                   IBB42610
       PZE                                                                   IBB42620
       PZE                                                                   IBB42630
       PZE                                                                   IBB42640
       PZE                                                                   IBB42650
       PZE                                                                   IBB42660
       PZE                                                                   IBB42670
       PZE                                                                   IBB42680
       PZE                                                                   IBB42690
       PZE                                                                   IBB42691
       PZE                                                                   IBB42692
       PZE                                                                   IBB42693
       PZE                                                                   IBB42694
       PZE                                                                   IBB42695
```

Figure 39. Auxiliary Disk/Drum Limits Table (SECTION 1)

```
ONLIN  BOOL    1                                                             IBD00210
DBLSPC BOOL    0                                                             IBD00220
KEYSWT SET     4                                                             IBD00230
FORMAT SET     3                                                             IBD00240
```

Figure 40. System Core-Storage Dump Assembly Parameters (SECTION 1)

```
NOCH   SET     0     NO. OF DIRECT ACCESS MODULES DEFINED IN IOCS *DKDM 11000310
NOHYP  SET     0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 11000330
```

Figure 41. IBJOB Monitor Assembly Parameters for Disk, Drum, and Hypertape (SECTION 2)

```
NOCH   EQU     0   NO. OF DIRECT ACCESS MODULES DEFINED IN IOCS  *DKDM  IOA00360
NOHYP  EQU     0     NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS   *HYPR  IOA00380
```

Figure 42. Full IOCS (Independent IOCS) Assembly Parameters for Disk, Drum, and Hypertape (SECTION 3)

```
NOCH   SET     0     NO. OF DIRECT ACCESS MODULES DEFINED IN IOCS *DKDM 3AE00060
NOHYP  SET     0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 3AE00080
NOCH   SET     0     NO. OF DIRECT ACCESS MODULES DEFINED IN IOCS *DKDM 3AF00270
NOHYP  SET     0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 3AF00290
NOCH   SET     0     NO. OF DIRECT ACCESS MODULES DEFINED IN IOCS *DKDM 3I000300
NOHYP  SET     0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 3I000320
```

Figure 43. IBJOB Subroutine Library Assembly Parameters for Disk, Drum, and Hypertape (SECTION 4)

0, Module 1, Data Channel Switch setting 1) as SYSUT3, the following cards would be inserted among the SECTION 1 PARAMETER cards in the update-edit deck:

```
D          BOOL     23101              IBB42449
SYSUT3     PZE      3,,D               IBB42450
```

When a direct access storage module is assigned to a system unit function, a matching entry must be made in the Auxiliary Disk/Drum Limits Table (Figure 39) which defines the limits of the disk or drum storage area that is to be assigned to the function. An entry in the Auxiliary Disk/Drum Limits Table has the following format:

```
           PZE      dorg,,dend
```

where dorg and dend are the numbers of the first and last tracks, respectively, of the consecutive tracks in the disk or drum storage module (which is specified in the corresponding entry in the Auxiliary System Unit Function Table) that are to be assigned to the system unit function. For example, if tracks 0040 through 0079 of the disk storage module in the previous example are assigned as SYSUT3, the following card would be inserted in the Disk/Drum/Hypertape Update-Edit Deck:

```
           PZE      40,,79             IBB42680
```

Inserting System Core-Storage Dump Assembly Parameters

If any changes are to be made in the distributed version of the assembly parameters for the System Core-Storage Dump Program (shown in Figure 40), the appropriate cards should be punched and inserted as part of the SECTION 1 PARAMETER cards in the Disk/Drum/Hypertape Update-Edit Deck, immediately following the cards for changing entries in the Auxiliary System Unit Function and Disk/Drum Limits Tables. The core-storage dump parameters are described in Figure 35.

Inserting IBJOB Monitor Assembly Parameters for Disk, Drum, and Hypertape Capability

The IBJOB Monitor assembly parameters for disk, drum, and Hypertape capability are listed in Figure 41. If only 1301 and 7320 capability is to be added, the card containing *DKDM in columns 67 through 71 should be punched and inserted as the SECTION 2 PARAMETER cards in the Disk/Drum/Hypertape Update-Edit Deck. The variable field of the card should contain a decimal number which indicates the number of 1301 and 7320 modules that may be used by the Component IOCS in the IBJOB Monitor. If only 7340 capability is to be added, the card containing *HYPR in columns 67 through 71 should be punched to indicate the number of 7340 Hypertape channels to be used by Component IOCS and inserted as the SECTION 2 PARAMETER cards in the Disk/Drum/Hypertape Update-Edit Deck. If 1301, 7320, and 7340 capability are all to be added, both cards should be prepared and inserted in SECTION 2 of the update-edit deck.

Inserting Full IOCS (Independent IOCS) Assembly Parameters for Disk, Drum, and Hypertape Capability

The Full IOCS (Independent IOCS) assembly parameters for direct access storage and Hypertape capability are listed in Figure 42. These parameters are inserted as the SECTION 3 PARAMETER cards in the Disk/Drum/Hypertape Update-Edit Deck and indicate the number of 1301 Disk Storage modules, 7320 Drum Storage modules, and/or 7340 Hypertape channels that may be used by the Full IOCS (Independent IOCS). Otherwise, the procedure for preparing and inserting these parameters is the same as for the IBJOB Monitor assembly parameters.

Inserting IBJOB Subroutine Library Assembly Parameters for Disk, Drum, and Hypertape Capability

The IBJOB Subroutine Library assembly parameters for direct access storage and Hypertape capability, shown in Figure 43, indicate the number of 1301 modules, 7320 modules, and/or 7340 Hypertape channels that may be used by the Library IOCS in the IBJOB Processor. The first four of these parameter cards are inserted as the SECTION 4 PARAMETER cards in the Disk/Drum/Hypertape Update-Edit Deck. The last two parameter cards are inserted, as indicated by their serial numbers, after the fifth card following the SECTION 4 PARAMETER insertion point of the update-edit deck. Otherwise, the procedure for preparing and inserting these parameters is the same as for the IBJOB Monitor.

Operating Procedure for the Disk/Drum/Hypertape Update-Edit Deck

After the Disk/Drum/Hypertape Update-Edit Deck has been prepared, as described in the preceding section, the following procedure should be followed to generate a new System Library Tape with disk, drum, and/or Hypertape capability.

1. Place the prepared update-edit deck in the System Input File on SYSIN1 (A2).
2. If disk and/or Hypertape capability is being added, mount the distributed 729-capability System Library Tape on SYSLB1 (A1). If drum capability is being added, either alone or along with disk and/or Hypertape capability, mount the System Library Tape which was obtained by running the symbolic tape with the symbolic modification deck (see the third paragraph under the "Introduction" to this section) on SYSLB1.
3. Mount the distributed IBSYS Symbolic Tape Number 1 on A3.

4. Mount work tapes on B1, B2, B3, B4, and A4.

5. Follow the initial start procedure described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

6. Follow the operating instructions printed on the System Printer.

The output tapes produced during the job run are described in the section "The Assignment and Function of Units for the Disk/Drum/Hypertape Update-Edit Deck." Refer to the "System Editor" section of the manual for information on editing the new System Library onto direct access storage or Hypertape.

Reassembling the System Monitor

Although reassembly of the System Monitor is normally not required unless direct access storage and/or Hypertape capability is to be added, it may be necessary at some installations to add an installation accounting routine or otherwise tailor the System Library to special installation requirements. The sample job deck in Figure 44 may be used for this purpose. Before the deck is used, any symbolic cards for inserting or changing coding, changing assembly parameters, and/or changing system unit assignments must be inserted in the deck immediately following the UPDATE pseudo-operation card. The assembly parameter and system unit assignment cards should be inserted in the order in which they appear in Figures 36 through 40. The System Monitor assembly parameters are described in Figure 35. The formats of the entries in the Auxiliary System Unit Function Table and Auxiliary Disk/Drum Limits Table are described in the preceding text, in the section "Inserting Changes to Entries in the Auxiliary System Unit Function and Disk/Drum Limits Tables."

For the sample job deck, the symbolic tape containing the System Monitor is mounted on A4 and the distributed System Library Tape is mounted on A1. The deck produces an updated symbolic tape containing the System Monitor on B3 and the new System Library Tape on A3.

Incorporating a User-Designed Installation Accounting Routine

A user may, if he wishes, design an installation accounting routine tailored to his requirements and incorporate it into the System Library as part of the 7090/7094 IBSYS Operating System. The accounting routine may be designed to perform a variety of functions, such as job timing, job billing, or produc-

```
$JOB          UPDATE AND ASSEMBLE SYSTEM MONITOR
$IBSYS
$ATTACH       A4
$AS           SYSCK1          UPDATE INPUT TAPE
$ATTACH       B3
$AS           SYSCK2          UPDATE OUTPUT TAPE
$REWIND       SYSCK1
$REWIND       SYSCK2
$EXECUTE      UPDATE
       UPDATE  9,10
   (INSERT ASSEMBLY PARAMETER CARDS, FIGURES 36 THROUGH 40)
       END                                                          IBD17690
       UNLOAD  9
$IBSYS                                                              IBD99998
$STOP                                                               IBD99999
       ENDFIL  10
       REWIND  10
       ENDUP
$IBSYS
$REWIND       SYSPP1
$*     PLEASE REMOVE UPDATE INPUT TAPE FROM UNIT A4. MOUNT A SCRATCH
$*     TAPE IN ITS PLACE.    THEN PRESS START TO CONTINUE.
$PAUSE
$EXECUTE      IBJOB
$IBJOB IBSYS  NOGO
$IEDIT        SYSCK2,SRCH
$IBMAP IBSYS  8500,ABSMOD,()OK
(END-OF-FILE CARD)
$IBSYS
$REWIND       SYSPP1
$REMOVE       SYSCK2
$ATTACH       B2
$AS           SYSUT2
$IBEDT
      *EDIT   MAP,MODS
TAPE  *REPLACE IBSYS
TAPE  *REPLACE SYSDMP
(END-OF-FILE CARD)
$IBSYS
$RESTORE
$STOP
```

Figure 44. Sample Job Deck for Updating and Assembling the System Monitor

ing statistical data on the processing of jobs. Since accounting practices vary considerably from installation to installation, a specific accounting routine is not provided with the 7090/7094 IBSYS Operating System. However, facilities are provided for incorporating into the system an installation accounting routine specifically designed by the user to meet his own requirements.

There are three one-word entries in the Communications Region of the System Nucleus that are used specifically for accounting purposes: SYSIDR, SYSACC, and SYSPID. The function of each of these entries is described in detail in the following text.

An installation accounting routine may either be incorporated as part of the System Monitor and be available for use by the System Monitor and each of the subsystems, or it may be incorporated as part of a subsystem where it is available for use by the subsystem only. Both methods of incorporating an accounting routine are described below.

Designing an Installation Accounting Routine

There are several considerations about the relationship of an installation accounting routine to the IBSYS operating system which should be kept in mind when designing an accounting routine. These considerations are discussed below.

Location and Size

An accounting routine must be loaded into upper core storage and must not occupy more than 500 words. There are two subsystems that will not respect the upper 500 words of core storage and may overlay all or part of an accounting routine.

1. FORTRAN II respects only the upper 64 words of core storage.

2. In IBJOB, if $DUMP or $PATCH cards are used, the JDUMP routine is loaded into upper core storage overlaying the accounting routine, and the contents of SYSIDR are replaced by TRA 2,4.

Calling Sequence

Access to an accounting routine is through location SYSIDR in the Communication Region of the System Nucleus. SYSIDR must be patched to contain a transfer to the accounting routine (see the section "Incorporating an Accounting Routine into the System Monitor"). Whenever the System Monitor or any of its subsystems processes a $JOB or a $ID card, a transfer is made to the accounting routine through SYSIDR (i.e., TSX SYSIDR,4). In addition, some processors under control of IBJOB, Commercial Translator, and 9PAC make a sign-on transfer to the accounting routine before they begin processing. There may be more than one sign-on for a job under IBJOB or Commercial Translator. For example, IBMAP will sign-on before starting assembly of a source program in the MAP language, and the Loader will sign-on before loading the object program assembled by IBMAP. When all processing is complete for a job under IBJOB or Commercial Translator, there will be one (and only one) sign-off transfer to the accounting routine.

The calling sequence used in transferring to the accounting routine is:

```
        TSX     SYSIDR,4
        pfx     loc,t,n
                return
```

where:

pfx = PZE for $JOB/$ID calls
PON for sign-on calls
PTW for sign-off calls
PTH for intraprocessor calls
(prefix codes FOR, FIVE, SIX, and SVN may be used for special purposes by an installation)

loc = location of first word of an identifying message or a control card buffer (not always present)

t = identifier of sign-on and sign-off calls
1 = compiler
2 = assembler
3 = loader
4 = execution
5 = utility
(6 and 7 may be used for special purposes by an installation)

n = number of words occupied by identifying message or control card buffer (not always present)

In some cases the sign of the AC is tested upon return from the accounting routine. If the sign is found to be plus (+), processing continues. If the sign is found to be minus (−), processing is terminated and control is returned to the supervising monitor. Figure 45 gives the contents of the parameter words for the calls by various processors and indicates whether the sign of the AC is tested on return (the sign of the AC is never tested on return from a sign-off call).

$JOB and $ID Cards

The contents of the $JOB or $ID card are stored in a buffer prior to being printed on-line. The location and length of this buffer is stored in the parameter word (PZE) of the calling sequence (see the section "Calling Sequence") so that the accounting routine may have access to the data in columns 16-72 of the $JOB card or columns 7-72 of the $ID card. The data on these cards may be set up in any form desired for use by the accounting routine.

SYSACC

The System Monitor or the subsystem processing a $JOB or $ID card tests location SYSACC. If the contents of SYSACC are ZERO, the card is printed on-line. If SYSACC contains anything other than zero, the card is not

Processor	$JOB/$ID Parameter Word*	$JOB/$ID Sign of AC Checked	Sign-On Parameter Word	Sign-On Sign of AC Checked	Sign-Off Parameter Word
IBSYS Monitor	PZE loc,,n	No			
Symbolic Update	PZE loc,,n	Yes			
Sort	PZE loc,,n	No			
IOCS	PZE loc,,n	No			
Utilities	PZE loc,,n	No			
IBJOB	PZE loc,,n	Yes			PTW 0,0,0
IBFTC			PON 0, 1, 0	**	
IBCBC			PON 0, 1, 0	No	
IBMAP			PON loc,2,n	Yes	
Loader	PZE loc,,n	Yes	PON loc,3,n	Yes	PTW loc,3,n
Execution					PTW 0,4,0
CT Monitor	PZE loc,,n	No			
Compiler			PON 0,1,0		PTW 0,1,0
Loader			PON 0,3,0	Yes	PTW 0,3,0
Postprocessor			PON loc,5,n	Yes	PTW loc,5,n
KPOUT***					PTH loc,,n
9PAC Monitor	PZE loc,,n	No			
Compiler			PON loc,1,n	No	
FORTRAN II	PZE loc,,n	Yes			

* n may be assumed to be 14 when it is omitted from the decrement field in $JOB/$ID calls.
** prints error message if AC is minus but does not discontinue processing.
*** KPOUT routine is entered when an error condition is encountered which necessitates discontinuing processing.

Figure 45. Installation Accounting Routine Calling Sequence Parameter Words

printed. In the distributed version of IBSYS, SYSACC contains:

PZE 0

If the accounting routine is to control printing of the $JOB and $ID cards, SYSACC must be patched with the following card:

1	8	16	73
SYSACC	PZE	any non-zero characters	IBB42020

If both SYSACC and SYSPID (see the section "SYSPID") are nonzero, the IBJOB Monitor prints a special page heading on each page of the job listing. The data for the special heading is contained in a 10-word buffer which the user must provide. The location of the first word of the buffer must be placed in the address portion of SYSACC. The 10-word buffer must be in the following form:

Words 1-2	Time of day
Words 3-4	Date
Words 5-7	Primary identification
Words 8-9	Secondary identification (e.g., charge number)
Word 10	Installation or run code

The routine in IBJOB that prints the special page heading is entered at symbolic location HED (1PC11060).

SYSPID

Location SYSPID in the System Nucleus is used only by the IBJOB Monitor. The IBJOB Monitor tests SYSPID only if SYSACC (see the section "SYSACC") is found to be nonzero. The status of SYSPID determines whether the standard or special page heading is to be printed on the job listing. If SYSPID is zero, the standard heading is printed. If SYSPID is nonzero, the special page heading is printed. The 10-word buffer described in the "SYSACC" section must be provided and a pointer to the buffer must be placed in SYSACC if SYSPID is nonzero. If location SYSPID is to be changed, it must be patched with a card serialized IBB42030.

Clock Reading Routine

If the accounting routine is to contain a clock reading routine, either of the following clocks may be used:

Core Storage Clock and Interval Timer – RPQF89349 (7090) or RPQ880295 (7094)
Program Accounting Clock – RPQ78054 (printer clock)

A routine which may be used to read the printer clock and print the reading on-line is included in Figure 46. If the programmer wishes to write his own printer clock reading routine, he must be careful not to use IOEX for this purpose because IOEX has no facilities for "reading" the on-line printer. He must also ensure that the channel to which the printer is attached is dormant before attempting to read the printer clock.

IOEX Message Writer

The IOEX Message Writer, .MWR, may be used to write accounting data on-line, off-line, or both (see the section "Message Writer"). Care must be exercised in using the Message Writer for writing data off-line. Off-line messages are processed by a subroutine called SPOUT. SPOUT is loaded into upper core storage from SYSEND-199 to SYSEND (see Figure 47). When an accounting routine is incorporated, SYSEND is moved back as shown in Figure 47. The Message Writer should not be

```
SYSEND EQU    -101                                              IBB00290
       MOVLDR ACTBEG,ACTEND,ACTORG MACRO TO MOVE ROUTINE TO UPPER CORE IBB07755
       MOVLDR ACTBEG,ACTEND,ACTORG MACRO TO MOVE ROUTINE TO UPPER CORE IBB34590
SYSIDR TRA    ACTBEG              LOCATION OF ACCOUNTING ROUTINE IBB41990
ACTORG BSS    100                 LOAD ROUTINE HERE FIRST        IBB47525
*             ACCOUNTING ROUTINE WHICH READS PRINTER CLOCK       IBB66820
*             CALLING SEQUENCE                                   IBB66830
*      TSX    SYSIDR,4                                           IBB66840
*      PFX    L,T,N                                              IBB66850
*                                                                IBB66860
       ORG    ACTORG                                             IBB66870
ACTBEG LOC    SYSEND+1                                           IBB66880
       SXA    CLOCK2,2            SAVE                           IBB66890
       SXA    CLOCK4,4            REGISTERS 2 AND 4              IBB66900
       AXC    0,1                 WAIT FOR                       IBB66910
       ZET*   .CHXAC              CHANNEL A DORMANT              IBB66920
       TRA    *-1                                                IBB66930
       TXI    *+1,1,-1            WAIT FOR                       IBB66940
       ZET*   .CHXAC              CHANNEL B DORMANT              IBB66950
       TRA    *-1                                                IBB66960
       AXT    25,4                                               IBB66970
       STZ    SCRACH                                             IBB66980
       RPRA                       START PRINTER IN MOTION        IBB66990
       SPRA   10                  TO READ QREAD OUTQ HUB OF CLOCK IBB67000
       RCHA   COMMD1              IN ORDER TO PRINT AND STORE CONTENTS,IBB67010
       LCHA   COMMD1              TOTAL OF 46 COPY MUST THEN FOLLOW. IBB67020
       TIX    *-1,4,1             IMAGEE IS THUS FILLED WITH 9 LEFT IBB67030
       LCHA   COMMD2              THROUGH 9RIGHT ECHO,WHILE A LINE OF IBB67040
       TCOA   *                   BLANKS AND TIME ARE PRINTED ON-LINE IBB67050
       AXT    18,4                IRX4 REFERS TO ROWS            IBB67060
NEXWRD AXT    36,2                9 LEFT, 8LEFT AND SO ON        IBB67070
       LDQ    IMAGEE+18,4         INTEROGATE THE ECHOS           IBB67080
NEXBIT TQP    NOBIT               IN IMAGEE THROUGH              IBB67090
       PXD    ,4                  IMAGEE+17 AND TRANSLATE        IBB97100
       ALS    11                  THE TIME THAT WAS REGISTERED   IBB67110
       ARS    36,2                INTO RESULT                    IBB67120
       ORS    SCRACH                                             IBB67130
NOBIT  RQL    1                   FINAL RESULT IS                IBB67140
       TIX    NEXBIT,2,6          BCD READING OF CLOCK           IBB67150
       TIX    NEXWRD,4,2          IN THE CELL SCRACH             IBB67160
       CLA    SCRACH                                             IBB67170
CLOCK2 AXT    **,2                RESTORE                        IBB67180
CLOCK4 AXT    **,4                REGISTERS 2 AND 4              IBB67190
       TRA    2,4                 RETURN TO MAIN PROGRAM         IBB67200
COMMD1 IOCT   SCRACH,,1                                          IBB67210
COMMD2 IOCP   IMAGEE,,2           READ 9 LEFT AND RIGHT ECHO     IBB67220
       IOCP   SCRACH,,1                                          IBB67230
       IOCP   SCRACH,,1                                          IBB67240
       IOCP   IMAGEE+2,16         READ 8 LEFT THROUGH 1 RIGHT ECHO IBB67250
SCRACH PZE                                                       IBB67260
IMAGEE BSS    18                                                 IBB67270
ACTEND BSS    0                                                  IBB67280
       ORG    SYSORG              NEW ORIGIN                     IBB67290
       TCD    *                                                  IBB67300
```

Figure 46. Sample Deck for Incorporating Installation Accounting Routine in System Monitor

used to write off-line accounting data for any subsystem which does not recognize SYSEND-200 as the end of usable core storage (e.g., IBJOB at object time).

Incorporating an Accounting Routine into the System Monitor

The process of incorporating an accounting routine in the System Monitor consists first of updating the Monitor and then reassembling it. The section "Reassembling the System Monitor" gives instructions for updating and reassembling the System Monitor.

There are two methods of incorporating an accounting routine into the System Monitor. In the first method, the routine is loaded directly into its permanent residence in upper core storage whenever the System Supervisor is loaded. In the second method, the routine is loaded first into lower core storage with the Supervisor and is only relocated to upper core storage at "cold start" time and when a subsystem returns control to the System Supervisor through location SYSRET. The details of these two methods and discussions of their respective advantages are given below.

Method 1

In this method the accounting routine is inserted into the System Supervisor with an ORG to SYSEND+1. The updating required to incorporate the accounting routine requires patching at three locations in the System Monitor as described below:

1. SYSEND is defined as -1 (i.e., 77777_8) in the distributed version of IBSYS. Since the accounting routine will occupy the final words of upper core storage, SYSEND must be redefined to the beginning of the routine minus 1. The patch card to redefine SYSEND must be in the following form:

1	8	16	73
SYSEND	EQU	$-(n+1)$	IBB00290

(where n is the number of instructions in the accounting routine.)

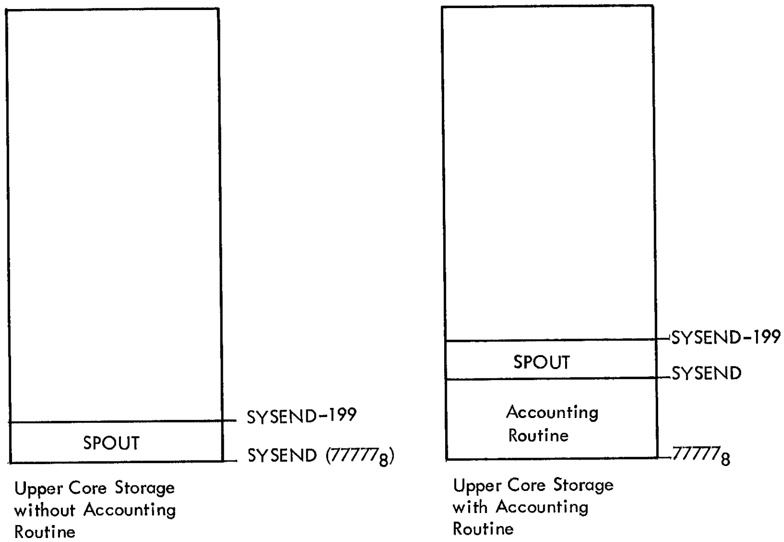


Figure 47. Relationship of Installation Accounting Routine to SPOUT Subroutine

2. SYSIDR contains the instruction TRA 2,4 in the distributed version of IBSYS. This instruction must be changed to a transfer to the accounting routine. The patch card to change this instruction must be in the following form:

1	8	16	73
SYSIDR	TRA	name	IBB41990

(where name is the name of the entry point of the accounting routine, i.e., SYSEND+1.)

3. The accounting routine is patched into the System Supervisor starting at serialization IBB66820, which is the highest serialization used in the Supervisor. In the distributed version of IBSYS, the card with this serialization contains the instruction TCD with an asterisk in the variable field. This instruction must be placed after the last instruction in the accounting routine. Its serialization must be the next highest in sequence after the last instruction of the accounting routine. The block of instructions to be patched in at IBB66820 is:

1	8	16	73
name	ORG	SYSEND+1	IBB66820
	.		
	. (installation accounting routine)		
	.		
	TCD	*	IBBxxxxx

(where IBBxxxxx is the next serial number in sequence after the last instruction of the accounting routine.)

The principal advantage of Method 1 over Method 2 (see the section "Method 2") is that Method 1 is somewhat simpler. The principal disadvantage of Method 1 is that each time the System Supervisor is called into core storage, the accounting routine is reloaded into upper core storage. This overlaying of the accounting routine with itself may cause difficulties in debugging. One such difficulty would occur if a subsystem overlaid part of the accounting routine. A dump would not reveal the overlaying because, when a call is made to the Core Storage Dump Program (SYSDMP), the System Supervisor is loaded into core storage before SYSDMP is loaded. Method 2 overcomes this disadvantage.

Method 2

In this method the accounting routine is inserted into the System Supervisor with an ORG to a predefined location in the System Supervisor so that whenever the Supervisor is loaded into core storage, the accounting routine is loaded into that predefined area. A subroutine, which the user must place at two locations in the System Supervisor, will move the accounting routine to permanent residence in upper core storage at "cold start" time and whenever a subsystem returns control to the System Supervisor through location SYSRET.

The updating required to incorporate the accounting routine requires patching at five locations in the System Monitor as described below:

1. SYSEND must be changed as described in Method 1.
2. A routine to move the accounting routine from the System Supervisor to upper core storage must be placed at two locations in the System Supervisor. First, a move routine must be placed in the COLD routine to move the accounting routine at "cold start" time. Second, a move routine must be placed in the TPPS routine, which is entered whenever control is returned from a subsystem to the System Supervisor through location SYSRET. This move routine will restore the

accounting routine if it has been overlaid by a subsystem or an object program. A special routine may be written and inserted, or use may be made of a macro-operation (MOVLDR) already available in the distributed version of IBSYS. To use this macro-operation insert the following card:

```
8        16                      73
MOVLDR   name1,name2,ACTORG      IBBxxxxx
```

(where name1 is the name of the location of the first instruction of the accounting routine after relocation, i.e., SYSEND+1; name2 is the name of the last instruction plus 1 in the accounting routine after relocation, i.e., 100000_8; ACTORG is the name of the first location into which the accounting routine is initially loaded, sec item 4 below; and IBBXXXXX is IBB07755 for the COLD routine and IBB34590 for the TPPS routine.) This macro-instruction will generate the following routine:

```
AXT      name2-name1,1
CAL      ACTORG+name2-name1,1
SLW      name2,1
TIX      *-2,1,1
```

3. SYSIDR must be patched as described in Method 1.

4. Storage allocation: The following card is included in the System Supervisor as distributed:

```
1        8        16                 73
ACTORG   BSS      0                  IBB47525
```

It is at this location that the accounting routine will be loaded initially and from which it will be relocated to upper core storage. This card must be patched with the following:

```
1        8        16                 73
ACTORG   BSS      n                  IBB47525
```

(where n is the number of instructions patched in at IBB66820; see item 5 below.)

5. The accounting routine is patched into the System Monitor in the same way as described in Method 1. The block of instructions to be patched in at IBB66820 for Method 2 is:

```
1        8        16                 73
         ORG      ACTORG             IBB66820
name1    LOC      SYSEND+1           IBB66830
  .
  . (installation accounting routine)
  .
         ORG      SYSORG             IBBxxxxx
         TCD      *                  IBBxxxxy
```

(where IBBxxxxx is the next serial number in sequence after the last instruction of the accounting routine and IBBxxxxy is the next serial number in sequence after IBBXXXXX.)

Figure 46 is a listing of a sample deck showing the five patches required for updating the System Monitor to incorporate an accounting routine by Method 2. The accounting routine shown (starting with ACTBEG+1 and ending with ACTEND) is not a complete installation accounting routine. It comprises only a tested clock reading routine that uses the printer clock.

The principal advantage of Method 2 over Method 1 is that the accounting routine is not overlaid before a SYSDMP dump is taken. The principal disadvantage is that the amount of upper core storage occupied by the System Supervisor is increased by the size of the accounting routine, and the time for executing Supervisor calls via SYSRET is increased by the time required to move the accounting routine.

Incorporating an Accounting Routine into a Subsystem

If an accounting routine is designed for use only with a specific subsystem, the accounting routine should be assembled and then appended to the first record of the subsystem by using an *MODIFY card when editing the System Library (refer to the section "System Editor"). This will result in the accounting routine being loaded whenever the first record of the subsystem is loaded, that is, after a $EXECUTE card containing the name of the subsystem is read. The first record of the subsystem should also be appended to overlay the SYSIDR location in the Communication Region of the System Nucleus so that the location contains a transfer to the beginning of the installation accounting routine. If necessary, the SYSCOR location should be overlaid in order to change the address portion of the entry which defines the end of usable core storage. The SYSACC location may also be overlaid if it is desired to control printing of the $JOB and $ID cards.

A $IBSYS card, followed by a $RESTORE card, should be placed at the end of a job or job segment performed by a subsystem containing an accounting routine if it is followed on the system input file by jobs or job segments to be performed by other subsystems or by the System Monitor. The $RESTORE card will restore SYSIDR, SYSCOR, SYSACC, and SYSPID locations to their original state, thereby nullifying any changes made by the subsystem containing the accounting routine.

Incorporating User Programs as Subsystems Under System Monitor Control

A user may, if he chooses, design a program and insert it in the System Library. The program can then be called into core storage by a $EXECUTE card and can be executed. Once the program is coded and assembled, it may be inserted in the System Library using a job deck similar to the sample job deck in Figure 48.

For the sample job deck, it is assumed that a user program, which is named SYSTMU, is located on SYSUT2 in the form of absolute column-binary card images that

```
1     7        16
--------------------------------------------------
$JOB           INSERT SYSTMU AS SUBSYSTEM AFTER IBJOB
$IBSYS
$IBEDT
      *EDIT    MAP,MODS
      *PLACE   SYSTMU,1,1,2
FILE  *AFTER   IBJOB
TAPE  *INSERT
   (COLUMN BINARY CARD IMAGES ON SYSUT2)
      *INSERT  FILEMK
   (END-OF-FILE CARD)
$STOP
```

Figure 48. Sample Job Deck for Inserting a User Program as a Subsystem Under Monitor Control

terminate with a transfer card. The job deck is designed to insert the program in the System Library immediately following the IBJOB Processor. Before the program is inserted, it is converted by the System Editor into a self-loading, scatter-load format, which is a standard format for the System Library. For the sample job deck in Figure 48, it is assumed that the user program consists of one record only. However, a program may consist of more than one record. For each additional record, another TAPE *INSERT card must be placed in the job deck and the absolute column-binary card images from which the record is formed must follow (on SYSUT2) the card images for the previous record. The column-binary card images for each record must end with a transfer card. Additional information may be found in the section "System Editor."

In designing and coding a program for insertion as a subsystem in the System Library, a number of rules must be adhered to in order to ensure proper loading of the program, coordinated control of input/output operations, and continuous job processing. These rules are as follows:

1. The program must use the core storage between SYSORG and SYSEND only (refer to SYSCOR in Appendix A).

2. The first word in the first record of the program must be a BCD name without leading blanks and may have an origin at SYSCUR (Appendix A). This name is the name of the program and is the name specified on the $EXECUTE card to call the program. The name must not be the same as the name of any other record in the System Library.

3. The first word of the second and succeeding records, if any, of the program must be a unique BCD name, without leading blanks, and may have an origin at SYSCUR.

4. The first record must contain a TRA instruction, which transfers control to the beginning of the program and which has an origin at SYSTRA (Appendix A).

5. The System Loader (SYSLDR) in the System Nucleus may be used by a program to load the second and succeeding records of the program. In Appendix A, a description of the use of the System Loader may be found under "SYSLDR." If the System Loader is used to load a record, the contents of SYSTRA must be changed to transfer control to the first instruction that is to be executed after the record is loaded. This may be accomplished by placing in the record itself a TRA instruction which has an origin at SYSTRA. The instruction will then overlay the content of SYSTRA when it is loaded.

6. The program must recognize and act upon the $IBSYS, $EXECUTE, $STOP, $ID, and $JOB cards, as described in the section "System Nucleus" under the heading "Job Control Communications with Subsystems."

7. If a system unit is to be used, it must be referred to by way of its entry in the System Unit Function Table and, if it is a direct access storage unit, by its entry in the Disk/Drum Limits Table (refer to the section "System Nucleus").

8. If a unit that is not assigned to a system unit function is to be used, it must be referred to by way of an availability chain (refer to the section "System Nucleus").

9. The program must either use IOEX (described in the section "Input/Output Executor") or adhere to the following rules:

   a. If a 729 Magnetic Tape Unit or 7340 Hypertape Drive is used, the program must keep track of the file and record count in the unit control block of the unit.
   b. Before using the System Loader, the program must make certain that there is no activity on the channel that is being used.
   c. If the System Loader is used to load records from direct access storage, the program must not overlay IOEX routines (FDAMT, (DECVD, and (DECRQ.

Incorporating IBM Modifications to the 7090/7094 IBSYS Operating System

The 7090/7094 IBSYS Operating System is distributed preassembled in binary form on a single 729 capability System Library Tape and in symbolic form on several tapes. Each of the symbolic tapes will normally contain more than one subsystem. However, since the System Monitor and each of the subsystems are maintained separately, instructions will be provided, when the symbolic tapes are distributed, for copying the System Monitor and each subsystem onto separate tapes. Whenever modifications are to be made to the System Monitor or a subsystem, a modification letter will be distributed. Two job decks will be distributed with the letter; a symbolic modification deck and a binary modification deck. Either deck may be used to produce a new binary System Library for 729 capability

with the required modifications incorporated. However, the symbolic modification deck will also produce an updated symbolic tape of the subsystem (or the System Monitor) being modified and a symbolic assembly listing of the portions of the subsystem that are updated and reassembled to incorporate the modifications. Those subsystems or components which have been reassembled for disk, drum and/or Hypertape capability can be maintained only with the symbolic modification deck issued with IBM modification letters. In these cases, the binary modification deck is not applicable because of address considerations. In addition to the symbolic and binary modification decks, a third job deck, the IBSYS Cumulative Editor Deck will be distributed whenever a new version of the 7090/7094 IBSYS Operating System is released. This deck may be used to accumulate modification cards from several binary modification decks and may be used, at any time, to bring a System Library up to the latest modification level in a single job run. Again, it cannot be used for modifying systems with disk/drum and/or Hypertape capability.

Any one or a combination of the three methods represented by the three job decks may be used at an installation to maintain the System Library, depending upon the particular requirements of the installation.

Symbolic Modification Deck

The symbolic modification deck distributed with each modification letter will be written in one of three standard forms. If the portion of the system being modified is written in the Macro Assembly Program (MAP) language (System Monitor, System Editor, IBJOB Processor, Utilities, or Restart Program), a deck similar to the one shown in Figure 49 will be used. In the case of modifications to the IBJOB Subroutine Library (IBLIB), a special symbolic deck will be distributed which is intended for use with this component of the IBJOB Processor only. A model of this deck will be found in Figure 50. If the portion of the system being updated is written in the FORTRAN II Assembly Program (FAP) language (Independent IOCS), a deck similar to the one shown in Figure 51 will be used. In all cases, the deck accompanying a modification letter will contain symbolic modification cards for updating a subsystem or the System Monitor in addition to the control cards listed in Figures 49, 50, and 51.

All three symbolic modification decks are designed for use with a System Library having the same system unit assignments (Figure 28) as the distributed System Library Tape. If system unit assignments and density settings have been changed at an installation, it may be necessary to change one or more of the $ATTACH

```
$JOB    MODIFICATION UPDATE-EDIT DECK FOR MAP ASSEMBLED SYSTEM
$IBSYS
$ATTACH         A4
$AS             SYSCK1          UPDATE INPUT TAPE
$ATTACH         B3
$AS             SYSCK2          UPDATE OUTPUT TAPE
$REWIND         SYSCK1
$REWIND         SYSCK2
$EXECUTE        UPDATE
        UPDATE  9,10
(SYMBOLIC UPDATE CARDS FOR MAP ASSEMBLIES TO BE MODIFIED)
        END                     END CARD OF LAST RECORD ON TAPE           XXXXXXXX
        UNLOAD  9

$IBSYS
$STOP
        ENDFIL  10
        REWIND  10
        ENDUP
$IBSYS
$REWIND         SYSPP1
$*      PLEASE REMOVE THE UPDATE INPUT TAPE FROM TAPE UNIT A4 AND MOUNT
$*      A SCRATCH TAPE IN ITS PLACE.    THEN PRESS START TO CONTINUE.
$PAUSE
$EXECUTE        IBJOB
$IBJOB IDNAME   NOGO
$IEDIT          SYSCK2,SRCH
(IBMAP CARDS FOR ASSEMBLIES TO BE MODIFIED)
(END-OF-FILE CARD)
$IBSYS
$REWIND         SYSPP1
$REMOVE         SYSCK2
$ATTACH         B2
$AS             SYSUT2
$IBEDT
      *EDIT     MAP,MODS
TAPE  *REPLACE (RECORD NAME)
(END-OF-FILE CARD)
$IBSYS
$RESTORE
$STOP
```

Figure 49. Symbolic Modification Deck for Updating MAP-Assembled System Components

```
$JOB    MODIFICATION UPDATE-EDIT DECK FOR IBJOB SUBROUTINE LIBRARY
$IBSYS
$REWIND         SYSUT1
$REWIND         SYSUT2
$REWIND         SYSUT3
$*      THE SYMBOLIC INPUT TAPE FOR THE IBJOB SUBROUTINE LIBRARY
$*      ON UNIT A4 WILL BE USED TO GENERATE THE NEW UPDATED
$*      SYMBOLIC TAPE ON UNIT B3.
$EXECUTE        UPDATE
        UPDATE  2,8
(SYMBOLIC UPDATE CARDS FOR LIBRARY ROUTINES TO BE ASSEMBLED)
        END                                                     3RN15370
        UNLOAD  2
$IBSYS                                                          3RN99998
$STOP                                                           3RN99999
        ENDFIL  8
        REWIND  8
        ENDUP
$IBSYS
$*      A SYSTEM INPUT TAPE FOR ASSEMBLING AND EDITING THOSE LIBRARY
$*      ROUTINES WHICH MUST BE ALTERED WILL NOW BE WRITTEN ON UNIT A3.
$EXECUTE        UPDATE
        UPDATE  8,4
$EXECUTE        IBJOB
$IBJOB IBLIB    MAP,LOGIC
$EDIT           LOGIC
        SKIPTO  (FIRST CARD OF FIRST ROUTINE TO BE ASSEMBLED)   AAAAAAAA
$REPLACE        (ROUTINE NAME)
        END     (END CARD OF FIRST ROUTINE ASSEMBLED)           BBBBBBBB
        SKIPTO  (FIRST CARD OF LAST ROUTINE TO BE ASSEMBLED)    CCCCCCCC
$REPLACE        (ROUTINE NAME)
        END     (END CARD OF LAST ROUTINE TO BE ASSEMBLED)      DDDDDDDD
        ENDFIL  4
        UNLOAD  8
        UPDATE  ,4,U
$IBSYS
$IBEDT
       *EDIT    MAP,MODS
FILE   *REPLACE CIFSR,SYSUT4
FILE   *REPLACE TIFSR,SYSUT4
        ENDFIL  4
$IBSYS
$STOP
        REWIND  4
        ENDUP
$IBSYS
$*      REMOVE SYMBOLIC UPDATE INPUT TAPE FROM TAPE UNIT A4.
$*      REMOVE SYMBOLIC UPDATE OUTPUT TAPE FROM TAPE UNIT B3.
$*      MOUNT SCRATCH TAPES ON BOTH THESE UNITS.
$*      SWITCH TAPE UNITS A2 AND A3.        PRESS START TO CONTINUE.
$PAUSE
```

Figure 50. Symbolic Modification Deck for Updating the IBJOB Subroutine Library

cards in the beginning of the deck accompanying a modification letter.

The assignment and function of tape units for the symbolic modification deck used for updating MAP coded portions of the system (excluding the IBJOB Subroutine Library) are shown in Figure 52. Those for the IBJOB Subroutine Library are shown in Figure 53 and those for the symbolic modification decks used for updating FAP coded portions of the system are shown in Figure 54.

The starting procedure for each of the three symbolic modification decks is the same. The old System Library Tape, containing the subsystem (or the System Monitor) to be modified, is mounted on SYSLB1 (tape unit A1). The old symbolic tape, with the subsystem (or the System Monitor) to be modified at the beginning of the tape, is mounted on SYSUT3 (tape unit A4). The symbolic modification deck is placed on the system input tape on SYSIN1 (tape unit A2). Follow the initial start procedure described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide,* Form C28-6355.

At the completion of any of the symbolic modification jobs, the new System Library Tape will be found on SYSUT1 (tape unit A3), the assembly listing of the updated and reassembled portions of the subsystem on SYSOU1 (tape unit B1), and the updated symbolic tape for the subsystem (or System Monitor) on SYSUT2 (tape unit B3). The density and unit function assignments for all units used by the symbolic modification decks are assumed to be as distributed (see Figure 28). If these system unit functions have not been modified, the physical tape units specified above will be valid for the functions designated, regardless of which form of the symbolic modification deck is used.

```
$JOB   MODIFICATION UPDATE-EDIT DECK FOR FAP ASSEMBLED SYSTEM
$IBSYS
$ATTACH        A4
$AS            SYSCK1         UPDATE INPUT TAPE
$ATTACH        B3
$AS            SYSCK2         UPDATE OUTPUT TAPE
$ATTACH        A3
$AS            SYSUT3         PROVIDES SYSUT3 FOR ASSEMBLY
$REWIND        SYSPP1
$REWIND        SYSCK1
$REWIND        SYSCK2
$EXECUTE       IBSFAP
      *FAP
       UPDATE  9,10,,D
       END                    SPACE TO BEGINNING OF PROPER ASSEMBLY      XXXXXXXX
       ENDUP
      *FAP
       UPDATE  9,10
(SYMBOLIC CHANGE CARDS TO BE UPDATED AND ASSEMBLED)
       END     -1                                                        YYYYYYYY
      *FAP
       UPDATE  9,10,,D
       END                    MATCH SERIALIZATION FOR LAST END CARD      ZZZZZZZZ
       ENDFIL
       ENDUP
$IBSYS
$SWITCH        SYSPP1,SYSUT2
$IBEDT
      *EDIT    MAP,MODS
TAPE  *REPLACE (RECORD NAME)
(END-OF-FILE CARD)
$IBSYS
$REMOVE        SYSCK1
$REMOVE        SYSCK2
$STOP
```

Figure 51. Symbolic Modification Deck for Updating FAP-Assembled System Components

Unit	Initial Assignment Function	Additional Assignments in this Job	Function in this Job	Update Logical Number
A1	SYSLB1	None	Old System Library	
A2	SYSIN1	None	System Input	
A3	SYSUT1	None	Intermediate New System Library	
A4	SYSUT3	SYSCK1	Old Symbolic Tape Intermediate	9
B1	SYSOU1	None	List Output	
B2	SYSPP1	SYSUT2	Punch Output	
B3	SYSUT2	SYSCK2	Updated Symbolic Tape Alternate Input Tape	
B4	SYSUT4	None	Intermediate	10

Figure 52. The Assignment and Function of Units for Symbolic Modification Decks Used for Updating MAP-Assembled Systems

The operations performed by each of the symbolic modification decks are listed below:

Operations Performed by Symbolic Modification Decks Used for Updating MAP Subsystems: The following operations are performed by the symbolic modification decks that are used to update portions of the system written in the Macro Assembly Program (MAP) language:

1. Tape unit A4 is attached as SYSCK1 to function as the update input tape.

Unit	Initial Assignment Function	Additional Assignment this Job	Function in this Job	Update Logical Number
A1	SYSLB1	None	Old System Library	
A2	SYSIN1	None	System Input	
A3	SYSUT1	None	Intermediate New System Library	4
A4	SYSUT3	SYSCK1	Old Symbolic Intermediate	9
B1	SYSOU1	None	List Output	
B2	SYSPP1	None	Punch Output	
B3	SYSUT2	SYSCK2	Updated Symbolic Intermediate	10
B4	SYSUT4	None	Intermediate	

Figure 53. The Assignment and Function of Units for Symbolic Modification Decks Used for Updating the IBJOB Subroutine Library

Unit	Initial Assignment	Additional Assignment	Function	Update Number
A1	SYSLB1	None	Old System	
A2	SYSIN1	None	System Input	
A3	SYSUT1	SYSUT3	New System	
A4	SYSUT3	SYSCK1	Old Symbolic	9
B1	SYSOU1	None	List Output	
B2	SYSPP1	SYSUT2	Punch Output	
B3	SYSUT2	SYSCK2 SYSPP1	New Symbolic	10
B4	SYSUT4	None	Intermediate	

Figure 54. Assignment and Function of Units for Symbolic Modification Decks for FAP-Assembled Systems

2. Tape unit B3 is attached as SYSCK2 to function as the update output tape.

3. The portion of the subsystem on the symbolic tape on SYSCK1 that requires modification is updated with the symbolic modification cards in the job deck on SYSIN1 and is written on SYSCK2. The unchanged portions are copied onto SYSCK2, after which a $STOP card with high serialization and an end-of-file mark are written.

4. The updated portions are assembled as an alternate input file on SYSCK2 by the MAP Assembler and written in binary form on SYSPP1. The assembly listing is placed on SYSOU1.

5. Tape unit B2 (SYSPP1) is attached as SYSUT2 in preparation for editing.

6. The System Library is edited to incorporate the updated and reassembled portions of the subsystem that are in binary form on SYSUT2. The following System Editor control card is normally used:

TAPE *REPLACE (record name)

In some cases, the following System Editor control card may be used:

TAPE *MODIFY (record name)

Operations Performed by Symbolic Modification Decks Used for Updating the IBJOB Subroutine Library: The following operations are performed by the symbolic modification deck that is used to update the IBJOB Subroutine Library:

1. Tape unit A4 is attached as SYSCK1 to function as the update input tape.

2. Tape unit B3 is attached as SYSCK2 to function as the update output tape.

3. The routines of the IBJOB Subroutine Library on the symbolic tape on SYSCK1 that require modifications are updated with the symbolic modification cards in the job deck on SYSIN1 and written on SYSCK2. The unchanged portions are copied onto SYSCK2, after which an end-of-file mark is written on SYSCK2.

4. A second and final update creating a system input tape is performed on the symbolic update output tape on SYSCK2. The Library subroutines which must be assembled are extracted and placed on SYSUT1, along with the proper control cards to perform a Librarian and IBSYS system edit to place the new IBJOB Subroutine Library on the System Library Tape.

5. Instructions for the operator are provided on the on-line printer to make the appropriate tape changes to allow the new system input tape prepared in phase 4 of the job to be run against the System Library Tape. Old and new symbolic tapes should now be removed and retained.

6. The appropriate routines are assembled by the IBJOB Processor, a Librarian edit is performed, and an IBSYS system edit is done from SYSUT4, producing a System Library Tape on SYSUT1 (tape unit A3) with a new IBJOB Subroutine Library. It should be noted that the symbolic output tape created in phase 3 of the job on SYSCK2 is the symbolic tape which will be used with future modifications to the IBJOB Subroutine Library.

Operations Performed by Symbolic Modification Decks Used for Updating FAP Subsystems: The following operations are performed by the symbolic modification decks that are used to update portions of the system written in the FORTRAN II Assembly Program (FAP) language:

1. Tape unit A4 is attached as SYSCK1 to function as the symbolic update input tape, and tape unit B3 is assigned to SYSCK2 to function as the symbolic update output tape.

2. The two functions SYSUT1 and SYSUT3 are assigned to tape unit A3. In addition to the normal list output on SYSOU1, the job deck produces a new System Library Tape and an updated symbolic tape of the subsystem being modified.

3. The System Peripheral Punch tape is rewound.

4. The symbolic tape of the subsystem, at the previous modification level, is copied from SYSCK1 onto SYSCK2 (without assembly) up to the portion that requires modification.

5. The portion of the subsystem on the symbolic tape on SYSCK1 that requires modification is updated with the symbolic modification cards in the job deck on SYSIN1 and is written on SYSCK2. The updated portion is assembled and then written in binary form on SYSPP1. The assembly listing is placed on SYSOU1.

6. The remainder of the old symbolic tape on SYSCK1 is copied (without assembly) onto SYSCK2, after which an end-of-file mark is written.

7. SYSPP1, which contains the updated and assembled portions of the subsystem, is switched with SYSUT2, in preparation for editing onto the new System Library Tape. The new symbolic tape is on the unit that is now assigned as SYSPP1, and may be saved.

8. The updated symbolic tape on SYSCK1 is rewound and unloaded.

9. The System Library is edited to incorporate the updated and reassembled portions of the subsystem that are in binary form on SYSUT2. The following System Editor control card is normally used:

TAPE *REPLACE (record name)

In some cases, the following System Editor control card may be used:

TAPE *MODIFY (record name)

Binary Modification Deck

The basic control cards in a standard binary modification deck are listed in Figure 55. In addition to these cards, each distributed deck will contain one or more

System Editor *MODIFY and/or *REPLACE cards and a number of column-binary alteration cards (Figure 18) for modifying a subsystem or the System Monitor. This deck can be used without change to produce a new System Library Tape from the System Library at the previous modification level. An *CHECK card is included in the deck to verify that the correct number of *MODIFY, *REPLACE, and column-binary alteration cards have been received.

The *MODIFY, *REPLACE, and column-binary alteration cards in the deck are serialized in columns 73 through 80. The serialization indicates the order, in the System Library, of the record being modified and has the following format:

AABCCDDD

where AA is the order in the System Library of the subsystem (or System Monitor) being modified, plus 10. For example, the System Monitor is 11, the IBJOB Processor is 12, the Symbolic Update Program is 13. B is the number of a file within the subsystem, CC is the number of a record within the file, and DDD is the number of a column-binary alteration card or an *MODIFY or *REPLACE card. For the *MODIFY and *REPLACE cards, DDD is always 000. As an example, the cards for modifying the first record (IBMAPJ) of the fourth file of the IBJOB Processor are serialized as follows:

*MODIFY IBMAPJ	12401000
(First alteration card)	12401001
(Second alteration card)	12401002
(Third alteration card)	12401003

The same type of serialization is used for the maintenance control cards in the System Editor portion of the symbolic modification deck shown in Figure 46.

IBSYS Cumulative Editor Deck

The basic control cards in the IBSYS Cumulative Editor Deck are listed in Figure 56. This deck is distributed whenever a new version of the 7090/7094 IBSYS Operating System is released. After each binary modification deck is received, modification cards from the deck may be removed and placed in the Cumulative Editor Deck. Instructions for doing this will be provided in each modification letter. The cumulative deck may be used to produce a System Library Tape at the current modification level from a backup System Library Tape. Use of the cumulative deck facilitates the determination of System Library modification levels, since the modification cards for the System Monitor and all subsystems are included in one deck.

Occasionally, it may be necessary to perform a special edit to modify library subroutines, using editing facilities in the IBJOB Processor, the FORTRAN II Processor, or the Commercial Translator Processor.

```
$JOB                                                                   00000001
$* IBJOB FORTRAN IV COMPILER (IBFTC)        7090-FO-805, VERSION 3, MOD. 2 00000008
$IBEDT                                                                 00000030
      *EDIT     MAP,MODS                                               00000031
       (MODIFICATION CARDS)                                            AABCCDDD
      *CHECK    N               VERIFY THAT N CARDS HAVE BEEN RECEIVED.  98999999
   (END-OF-FILE CARD)                                                  99000000
$STOP                                                                  99000001
```

Figure 55. Binary Modification Deck

```
$JOB            7090/7094 IBSYS, 7090-PR-130, CUMULATIVE EDITOR DECK     00000001
$*THIS EDITOR DECK WILL GENERATE A SYSTEM TAPE CONTAINING               00000002
$*  IBSYS SYSTEM MONITOR                  7090-SV-918, VERSION 6, MOD. 0 00000003
$*  IBJOB MONITOR (IBJOB)                 7090-SV-801, VERSION 5, MOD. 1 00000004
$*  IBJOB LOADER (IBLDR)                  7090-SV-802, VERSION 5, MOD. 1 00000005
$* +IBJOB SUBROUTINE LIBRARY (IBLIB)      7090-LM-803, VERSION 5, MOD. 0 00000006
$*  IBJOB MACRO ASSEMBLY PROGRAM (IBMAP)  7090-SP-804, VERSION 5, MOD. 0 00000007
$*  IBJOB FORTRAN IV COMPILER (IBFTC)     7090-FO-805, VERSION 5, MOD. 1 00000008
$*  IBJOB COBOL COMPILER (IBCBC)          7090-CB-806, VERSION 5, MOD. 0 00000009
$*  IBJOB DEBUGGING PROCESSOR (IBDBL)     7090-PR-807, VERSION 2, MOD. 0 00000010
$*  SYMBOLIC UPDATE PROGRAM (UPDATE)      7090-UT-978, VERSION 1, MOD. 0 00000011
$*  GENERALIZED SORTING SYSTEM            7090-SM-922, VERSION 7, MOD. 4 00000012
$* +FORTRAN II PROCESSOR                  7090-FO-928, VERSION 3, MOD.29 00000013
$*  UTILITIES                             7090-UT-927, VERSION 3, MOD. 0 00000014
$*  RESTART PROGRAM                       7090-IO-976, VERSION 2, MOD. 0 00000015
$* +COMMERCIAL TRANSLATOR PROCESSOR       7090-CT-921, VERSION 5, MOD. 6 00000016
$*  9PAC PROCESSOR                        7090-PR-924, VERSION 4, MOD. 2 00000017
$*  INPUT/OUTPUT CONTROL SYSTEM           7090-IO-919, VERSION 6, MOD. 2 00000018
**                                                                       00000019
$*       + MODIFICATION LETTERS TO THESE SYSTEMS WILL INDICATE WHETHER   00000020
$*         SPECIAL EDITS ARE REQUIRED TO BRING THE LIBRARIES OF THESE    00000021
$*         SYSTEMS TO THE MODIFICATION LEVELS STATED.                    00000022
$*                                                                       00000023
$IBEDT                                                                   00000030
      *EDIT     MAP,MODS                                                 00000031
       (MODIFICATION CARDS)                                              AABCCDDD
       END-OF-FILE CARD                                                  99000000
$STOP                                                                    99000001
```

Figure 56. IBSYS Cumulative Editor Deck

When this is necessary, a special modification deck is distributed. However, the cards from these special decks should not be incorporated in the Cumulative Editor Deck. In particular, see the first modification letter issued to the IBJOB Subroutine Library after the release of a new version for details concerning the IBLIB Cumulative Editor Deck.

The user may keep a copy of the distributed System Library Tape as a backup tape and use it and the Cumulative Editor Deck whenever it is necessary to produce a System Library Tape at the current modification level. If a special modification deck is distributed, it may be used to produce a new backup System Library Tape from the old one. The new backup tape and the Cumulative Editor Deck may then be used, when necessary, to produce an up-to-date System Library Tape.

Maintaining a Two-Tape System Library

Some installations may use a two-tape System Library, in which parts of the IBJOB Processor are located on a second System Library Tape. To incorporate IBM modifications without changing the distributed modification decks, a duplicate of the System Library should be maintained on a single tape reel. After modifications are incorporated in the single System Library Tape, it may be used to produce a two-tape System Library, in which parts of the IBJOB Processor are located on a second tape. A discussion of the use of multiple library units is contained in the publication *IBM 7090/7094 IBSYS Operating System: IBJOB Processor*, Form C28-6389.

7320 Drum Update-Edit Decks for IBJOB

The 7320 Drum Update-Edit Decks are designed to serve as a guide for providing IBJOB system residence on 7320 Drum Storage. These decks are shown in Figures 57 and 58.

Edit Deck Number 1 splits the IBSYS Operating System, on a 729 Disk/Drum/Hypertape-Capability Sys-

```
$JOB           7320 EDIT DECK NO. 1
$ATTACH        CNAM/S
$AS            SYSUT1,010,000
$ATTACH        A4
$AS            SYSUT3,H
$IBEDT
        *EDIT     MAP,MODS,CNAM/S
        *PLACE    UPDATE
        *PLACE    SORT
        *PLACE    IBSFAP
        *PLACE    FORTRA
        *PLACE    DK9OUT
        *PLACE    RESTAR
        *PLACE    CT
        *PLACE    9PAC
        *PLACE    IOCS
        *PLACE    CIFSR,2,2,1
        *PLACE    UPDATE,1,2,2
        *PLACE    SORT,2,2,3
        *PLACE    IBSFAP,1,2,4
        *PLACE    FORTRA,5,2,5
        *PLACE    DK9OUT,1,2,6
        *PLACE    RESTAR,1,2,7
        *PLACE    CT,10,2,8
        *PLACE    9PAC,4,2,9
        *PLACE    IOCS,3,2,10
        *MODIFY   IBSYS
15141   *OCT      400001000001             A1 IS SYSLB2
        *MODIFY   IBJOB                    CHANGE ACTION LIST FOR IBLIB
11117   *OCT      200000000000
11122   *OCT      200001000000
21335   *OCT      200000000000
        *AFTER    IBLDRO
        *DUP      SYSLB1,SYSALT,1          COPY OFF CIFSR
        *INSERT   FILEMK
        *AFTER    IBLDRQ
        *DUP      SYSLB1,SYSALT,1          COPY OFF TIFSR
        *INSERT   FILEMK
        *AFTER    IBLDRS
        *AFTER    FILEMK
        *DUP      SYSLB1,SYSALT,28         COPY OFF REST OF SYSTEMS
         END-OF-FILE CARD
$IBSYS
$PAUSE             CHANGE A4 TO A1, CLEAR CORE,LOAD DISK LOAD CARD.
```

Figure 57. 7320 Update-Edit Deck Number 1

```
$JOB  .        7320 EDIT DECK NO. 2
$ATTACH        A3
$AS    .       SYSCK1
$ATTACH        B3
$AS            SYSCK2
$REWIND        SYSCK1
$REWIND        SYSCK2
$REWIND        SYSPP1
$EXECUTE       UPDATE
       UPDATE  9,10
DRUM   EQU     1                                                   2M012200
DRUM   EQU     1                                                   2O012200
DRUM   EQU     1                                                   2P012200
       CAL*    UIIC41                PATCH FOR VERSION 5        (1)2P462800
DRUM   EQU     1                                                   2Q012200
DRUM   EQU     1                                                   2S012200
       END     -1                                                  2S530500
       ENDFIL  10
       REWIND  10
       UNLOAD  9
       ENDUP
$IBSYS
$PAUSE         PLEASE MOUNT SCRATCH TAPE ON A3.
$EXECUTE       IBJOB
$IBJOB IBLDR   NOGO
$IEDIT         SYSCK2,SRCH
$IBMAP IBLDRM  8000,ABSMOD,JOBSYM,()OK                             2M000040
$IBMAP IBLDRO  4000,ABSMOD,JOBSYM,()OK                             2O000100
$IBMAP IBLDRP  6000,ABSMOD,JOBSYM,()OK                             2P000100
$IBMAP IBLDRQ  5000,ABSMOD,JOBSYM,()OK                             2Q000100
$IBMAP IBLDRS  6000,ABSMOD,JOBSYM,()OK                             2S000100
'      END-OF-FILE CARD
$IBSYS
$ENDFILE       SYSPP1
$REWIND        SYSPP1
$REMOVE        SYSCK2
$PAUSE         PLEASE MOUNT SCRATCH TAPE ON B3.
$SWITCH        SYSPP1,SYSUT2
$SWITCH        SYSUT3,SYSUT1
$*     A3 WILL BECOME FINAL SYSTEM TAPE.
$IBEDT
      *EDIT    MAP,MODS
TAPE  *REPLACE IBLDRM
TAPE  *REPLACE IBLDRN
TAPE  *REPLACE IBLDRO
TAPE  *REPLACE IBLDRP
TAPE  *REPLACE IBLDRQ
TAPE  *REPLACE IBLDRS
'      END-OF-FILE CARD
$IBSYS
$REMOVE        SYSLB1
$ATTACH        A4
$AS            SYSLB1
$ATTACH        A3
$AS            SYSUT1
$ATTACH        B3
$AS            SYSUT3
$EXECUTE       IBJOB
$IBJOB         LOGIC
$EDIT          LOGIC
'      END-OF-FILE CARD
$IBSYS
$IBEDT
      *EDIT    MAP,MODS
FILE  *REPLACE CIFSR,SYSUT4
FILE  *REPLACE TIFSR,SYSUT4
'      END-OF-FILE CARD
$STOP
```

Figure 58. 7320 Update-Edit Deck Number 2

tem Library Tape, between a 7320 drum and a 729 System Tape. The IBSYS System Monitor, the IBJOB Monitor (minus the IBJOB Subroutine Library (IBLIB)), and the IBSYS Editor are edited onto a 7320 drum, while the IBJOB Subroutine Library and all other subsystems under the IBSYS Operating System are edited onto a 729 tape. This 729 System Library Tape (SYSLB2) is used in conjunction with those systems edited onto the 7320 drum. Edit Deck Number 1 accomplishes this split of the IBSYS Operating System with no assembly or reblocking of components required.

Octal modification cards are included in deck number 1 for the IBSYS and IBJOB records to modify the system to do the following:

1. Attach tape unit A1 as SYSLB2.

2. Modify the IBJOB ACTION table to indicate that IBLIB is on SYSLB2.

A one-pass IBEDT is performed. During this edit, SYSLB2 is created on tape unit A4. A map of the system as it resides on 7320 drum and on a 729 tape is written on the system output tape. At the end of the

edit run SYSLB2 (on tape-unit A4) should be manually switched to tape unit A1. A drum load card, punched on-line during the course of the job, should be used to initiate the operation of the split system.

Figure 59 is a summary of the input/output assignment for Edit Deck Number 1.

UNIT	ASSIGNMENT	FUNCTION
A1	SYSLB1	729/1301/7320 System Library (see Note)
A2	SYSIN1	System Input
A3	Available	Not Used
A4	SYSUT3	New System Library (SYSLB2)
B1	SYSOU1	List Output
B2	SYSPP1	Not Used
B3	SYSUT2	Intermediate
B4	SYSUT4	Not Used
cNam/s	SYSUT1	New System Library (SYSLB1)

(where c = channel, N = NEED DRUM, a = access arm, m = module, s = data channel switch)

NOTE: The System Library must contain the IBSYS modifications for the installation 7909 channel configuration. In particular the IBSYS record must contain the parameters specifying the 7320 drum configuration.

Figure 59. Summary of Input/Output Assignments for the 7320 Edit Deck Number 1

If an installation has two or more 7320 drums, the 7320 Update-Edit Deck Number 2 may be used to reassemble IBLDR and reblock IBLIB to a block size of 524 words to allow full track IBLIB residence on drum. Library records of 524 words will require approximately 12 less drum tracks for IBLIB residence and is only required if space on a drum is limited. A single system tape is produced as a result of applying this deck to a previously prepared System Library Tape with disk, drum, and/or Hypertape capability. An individual user may arrange the components on drums as desired, by preparing his own edit deck.

Symbolic modifications are included in IBLDR to allow full track residence of IBLIB on 7320 drum. The Update-Edit Deck Number 2 operates in five phases:

Phase 1	Updates IBLDR with the correct parameters to process subroutine blocks of 524 words when assembled.
Phase 2	Assembles the updated IBLDR.
Phase 3	Generates an intermediate System Library Tape with the reassembled IBLDR replacing the distributed IBLDR.
Phase 4	Reblocks IBLIB to a block size of 524 words.
Phase 5	Generates final System Library Tape by replacing distributed IBLIB with reblocked IBLIB through an IBSYS edit.

Following both phases 1 and 2 a stop occurs, at which time the symbolic input tape and symbolic output tape for IBLDR are removed from tape units A3 and B3 respectively and replaced by work tapes.

At the end of the run, the new System Library, SYSLB1, is on tape unit A3. Punch output of the assembly of IBLDR is destroyed by phase 3. If this output is desired, a pause should be inserted in the deck following phase 2, and the tape on B2 should be removed and replaced with a work tape. Figure 60 gives a summary of input/output assignments for Edit Deck Number 2.

	Assignment					
Unit	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Function
A1	SYSLB1	SYSLB1	SYSLB1	------	------	System Library (See Note)
A2	------	------	SYSIN1	------	------	System Input
A3	SYSUT1	SYSUT1	SYSUT3	SYSUT1	SYSUT1	Phase 1 - Update Input Phase 4 - New System Library
A4	SYSUT3	SYSUT3	SYSUT1	SYSLB1	SYSLB1	Phase 3 - Intermediate System Library
B1	------	------	SYSOU1	------	------	List Output
B2	SYSPP1	SYSPP1	SYSUT2	SYSUT2	SYSUT2	Intermediate
B3	SYSUT2 SYSCK2	SYSUT2 SYSCK2	SYSPP1	SYSUT3	SYSUT3	Intermediate
B4	SYSUT4	SYSUT4	SYSUT4	SYSUT4	SYSUT4	Intermediate

NOTE: The System Library must contain the IBSYS modifications for the installation 7909 channel configuration. In particular the IBSYS record must contain the parameters specifying the 7320 drum configuration.

Figure 60. Summary of Input/Output Assignments for the 7320 Edit Deck Number 2

Appendix A: System Nucleus Communication Region Entries

The functions of the entries in the Communication Region of the System Nucleus are described in this appendix.

SYSTRA is loaded with a transfer instruction when the first record of a subsystem is scatter-loaded into core storage following the reading of a $EXECUTE card. After the loading of the first record of a subsystem is completed, the System Loader transfers control to this entry, which, in turn, transfers control to the beginning of the subsystem. A transfer instruction may also be loaded in this location when succeeding records of a subsystem are loaded.

SYSDAT contains a six-character word containing the date specified on the last $DATE card processed. The entry should be updated by a $DATE card at the beginning of each day. The date word is provided for use in headings and labels by subsystems and object programs.

SYSCUR contains the name, in BCD form, of the subsystem or subsystem record currently in core storage. The System Supervisor places the subsystem name from the $EXECUTE card in this location before loading the first record of the subsystem. When succeeding records of a subsystem are loaded, the name of the record may be loaded into this location.

SYSRET is the location to which each subsystem transfers control to call in the System Supervisor. Once the System Supervisor is called into core storage, the instruction EMTM is executed. Therefore, the machine is always in the multiple-tag mode when control is passed to a subsystem from the System Supervisor.

SYSKEY is the location in which the entry key settings are stored at initial start. This location may be interrogated to determine what the status of the entry keys was before initial start.

SYSSWS is the location in which sense switch settings are stored at initial start. Sense switches 1 through 4 are reserved for use by the Operating System. This location may be interrogated to determine what the status of the sense switches was before initial start. Bits 30 through 35 of the entry represent sense switches 6 through 1, respectively. A 1-bit indicates that the corresponding sense switch was down at initial start.

SYSPOS contains the number of the System Library Unit and the position on the unit of the subsystem currently in core storage. This information is entered in the SYSPOS location by the System Supervisor after it looks up, in the System Name or System Loader Table, the position of a subsystem specified on the $EXECUTE card. When the System Library is on tape, the format of the entry is as follows:

```
PZE        INDEX,,NFILES
```

where INDEX is 1, 2, 3, or 4, corresponding to SYSLB1, 2, 3, or 4, and NFILES is the number of files the System Supervisor must skip over before loading the first record of the subsystem.

When the System Library is on disk or drum, the format of SYSPOS entry is as follows:

```
PZE        INDEX,,L(SYSNAM)
```

where INDEX is the same as previously described and L(SYSNAM) is the binary disk or drum track location of the subsystem.

SYSUNI contains the first location (in the address portion) and length (in the decrement portion) of the System Unit Function Table. The Disk/Drum Limits Table is assembled in the core-storage locations immediately following the System Unit Function Table.

SYSUBC contains the first location (in the address portion) and length (in the decrement portion) of the Unit Control Block Table. This table consists of one word for each channel containing the following information:

Prefix	Number of card units assigned to the channel.
Decrement	Total number of units assigned to the channel.
Address	Address of the first unit control block for the channel.

SYSUAV contains the first location (in the address portion) and length (in the decrement portion) of the Unit Availability Table. This table is described in the section "System Nucleus" under the heading "Unit Availability Table."

SYSUCW contains the first location (in the address portion) and the combined length (in the decrement) of all unit control blocks. The unit control blocks are assembled in contiguous locations.

SYSRPT contains a transfer instruction to a System Nucleus routine that detemines whether the System Supervisor or the subsystem receives control when a $JOB card is read by a subsystem. When a subsystem reads a $JOB card, it transfers to SYSRPT. If the sign of SYSJOB is minus, indicating that restoration of unit assignments is required, or if sense switch 1 is down and the card reader is not assigned as the System Input Unit, indicating that a between-jobs interrupt

condition exists, control is passed to the System Supervisor. Otherwise, control is returned to the subsystem that read the $JOB card.

SYSCEM normally contains the following instruction:

```
TRA     SYSTRA-2
```

This location is reserved for use by the customer engineer. During machine maintenance periods, it may contain a transfer to a customer engineering diagnostic routine located in core storage between SYSORG-50 and SYSORG-1. The diagnostic routine is transferred to this area by the System Supervisor at initial start and when a $RESTORE card is processed.

SYSDMP contains a transfer to a bootstrap routine for loading the System Core-Storage Dump Program, which is part of the System Monitor (IBSYS) file on SYSLB1. A transfer to SYSDMP initiates a core-storage dump in accordance with the options selected by the programmer or operator. (Additional information is contained in the section "System Core-Storage Dump Program.") The dump spill tape unit is SYSPP2. Neither SYSOU1 nor SYSPP2 can be disk or drum.

SYSIOX contains the first location (in the address portion) and length (in the decrement portion) of the IOEX Communication Table (Figure 16). This table contains entries for transferring control to IOEX subroutines.

SYSIDR is provided for transferring control to an installation accounting routine. Whenever a $ID or $JOB card is processed by a subsystem or the System Monitor, a transfer is made to SYSIDR, as follows:

```
TSX     SYSIDR,4
PZE     L($ID)
   return
```

where L($ID) is the location of the first word of the buffer containing the $ID or $JOB card in BCD form. In the distributed version of the System Monitor, SYSIDR contains:

```
TRA     2,4
```

Therefore, control is returned immediately to the subsystem (or to the System Monitor) that processed the $ID or $JOB card. If an installation accounting routine exists at an installation, SYSIDR should contain a transfer to the routine.

SYSCOR contains the limits of the core-storage area available for use by subsystems operating under control of the System Monitor.

```
PZE     SYSEND,,SYSORG
```

In the distributed version of the System Monitor, SYSEND and SYSORG are defined as follows:

SYSORG = 2652_8 or 1450_{10}
SYSEND = 77777_8 or $32{,}767_{10}$

The FORTRAN II Processor does not refer to location SYSCOR when defining the two symbols SYSORG and SYSEND. These symbols are defined by FORTRAN II as follows:

SYSORG = 3720_8 or 2000_{10}
SYSEND = 77777_8 or $32{,}767_{10}$

The user may redefine SYSEND to allow space for an installation accounting routine in upper core storage. In this case, the following limits apply: for FORTRAN II, SYSEND may not be lower than 77677_8 or $32{,}703_{10}$; for all other subsystems, SYSEND may not be lower than 77013_8 or $32{,}267_{10}$.

SYSLDR contains a transfer to the System Loader. The System Loader may be used to scatter-load subsystem records that have the same standard System Library record format as the first record of the subsystem. Each IOCP command used by the System Loader must have a word count no greater than $37777)_8$ and must be in the transmitting mode (bit 19 off). When loading is completed, the System Loader transfers to the SYSTRA location. Therefore, the contents of SYSTRA must be modified during or prior to loading.

Whenever the first record of a subsystem is loaded into core storage following the reading of a $EXECUTE card, the System Supervisor places in the decrement portion of SYSLDR the location of the unit control block for the unit from which the subsystem is being loaded. Subsequent transfers to SYSLDR by the subsystem will result in the next sequential record being loaded from the unit indicated in the decrement portion of SYSLDR. Therefore, if the decrement of SYSLDR is not changed by the subsystem, the next record of the subsystem will be loaded from the same unit as the previous record, each time a transfer is made to SYSLDR.

When the subsystem being loaded is on direct access storage, the System Loader routine obtains the track address of the next sequential record from the decrement of SYSTCH (described later in the text). When editing records onto disk or drum, the System Editor appends to each record a TCH SYSCYD command containing the track address of the next record in its decrement. When a record is loaded from direct access storage by the System Loader, the TCH command at the end of the record ends up in SYSTCH, where it is availabe for loading the next record.

A subsystem may specify the unit from which the next record is to be loaded by changing the unit control block address in the decrement portion of SYSLDR. However, the unit must be on the same channel (and Data Channel Switch setting if it is a 7909 Channel) as the unit from which the first record of the subsystem specified on the $EXECUTE card was loaded. When loading from direct access storage, the decrement of SYSTCH must also be changed.

The System Loader may be entered using the following instruction:

```
TSX    SYSLDR,4
```

If SYSLB1 is direct access storage and the System Supervisor that processed the last $EXECUTE card was loaded from direct access storage, an alternate entry may be made to the System Loader by using the following sequence:

```
TSX    SYSLDR,4,1
BCI    1,SYSREC
```

where SYSREC is the name of a record on SYSLB1. When this entry is made, the System Loader will load the record specified by SYSREC. The System Loader obtains the track address of the specified record from the System Loader Table (SLTABL). This table is generated and placed in the System Library when it is edited onto direct access storage by the System Editor. To obtain the track address, the System Loader writes a checkpoint record on SYSLB1 (just behind the System Core-Storage Dump Program), loads the System Loader Table into core storage, looks up the address of the specified record in the table, and restores core storage.

SYSACC is used for communication between the installation accounting routine (if one exists), and the subsystems and System Monitor. In the distributed version of the System Monitor, this location contains the following:

```
PZE    0,0,0
```

Whenever a $ID or $JOB card is processed by a subsystem or the System Monitor, SYSACC is tested before a transfer is made to SYSIDR. If SYSACC contains all zeros, the subsystem or System Monitor lists the $ID or $JOB card on the System Output and System Printer Units before transferring to SYSIDR. In the case of a $JOB card, a page eject is performed before the card is listed. If the contents of SYSACC are nonzero, the $ID or $JOB card is not listed before the subsystem or System Monitor transfers to SYSIDR. The installation accounting routine is provided with the location of the first word of the buffer containing the $ID or $JOB card (as described previously under SYSIDR) and must list the card if SYSACC is set to nonzero at an installation.

SYSPID is reserved for use in communication between an installation accounting routine, when one is incorporated in the IBSYS Operating System, and the subsystems and the System Monitor. The exact use of this location depends on the design of the installation accounting routine. In the distributed version of the Operating System, SYSPID is not used.

SYSCYD and SYSCYD+1 contain the following input/output commands, which are used by the System Loader when subsystem records are loaded from disk, drum, or Hypertape.

```
SYSCYD    CPYD    0,,0
          TCH     SYSTWT
```

SYSSLD, SYSTCH, and SYSTCH+1 contain the following input/output command sequence, which is used by the System Loader for scatter-loading subsystem records:

```
SYSSLD    CPYP    *+1,,1   (IOCP)
SYSTCH    WTR     *+1
          TCH     *-2
```

Upon completion of loading from direct access storage via the System Loader, SYSTCH will contain a TCH command whose decrement will contain the track address of the next sequential record on disk or drum.

SYSTWT contains a TWT instruction which serves as a common 7909 channel transfer point for all users of IOEX.

SYSGET contains a word that indicates to the System Supervisor why control was returned to it by a subsystem. Additional information on SYSGET is contained in the section "System Nucleus."

SYSJOB contains a control word that is used by the System Supervisor and the subsystems in controlling the skipping of jobs and job segments. Additional information on SYSJOB is contained in the section "System Nucleus."

CHEXI is the Direct-Couple-environment indicator location. An indirect zero test of this location (ZET* CHEXI) by a subsystem indicates the current operational mode of the system. A successful zero test indicates direct mode D-C environment.

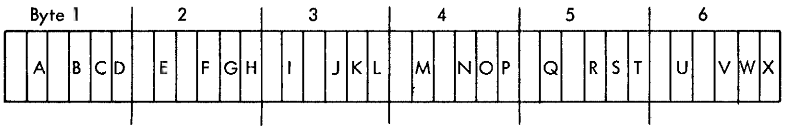
.MODSW is a Direct-Couple cell which indicates to IOEX whether the next record is in BCD or binary mode. If the contents are zero, binary mode is indicated; if nonzero, BCD is indicated.

Appendix B: Routine to Perform an IOEX Read or Write Using 729 Tape

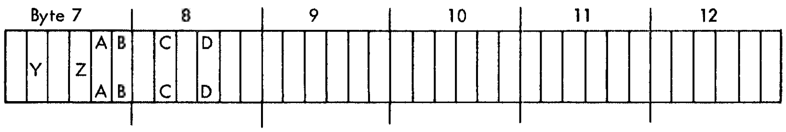
```
   THE FOLLOWING EXAMPLE IS SOLELY FOR THE PURPOSE OF ILLUSTRATION.
     IT DOES NOT REFLECT BUFFER TECHNIQUES IN USE OF 7607 CHANNEL.

*        SAMPLE ROUTINE   PERFORM READ OR WRITE FROM TAPE USING IOEX
*
*        CALLING SEQUENCE
*        TSX      IOXRW,4
*        P        LOCFIL,M,RCHSEQ
*        PZE      EOF,,ERR     EOF OR EOT OR ERROR RETURNS
*
*                 WHERE...
*
*        P        IF MINUS, WRITE   IF PLUS, READ
*                 BIT 1   IF 1, NO MESSAGE   IF 0, MESSAGE
$        LOCFILE LOCATION OF A WORD WITH UCB IN ADDRESS  (INDIR. REF.)
*        M        IF BCD,1   IF BINARY, 0
*        RCHSEQ   LOCATION OF I/O COMMANDS (ENDING IN TRAP)
*        EOF      END OF FILE OR TAPE EXIT
*        ERR      ERROR EXIT
                  INDICATORS AND ACC ARE DESTROYED.  IRS ARE SAVED.
*
*
IOXRW    SXA      IOXS4,4          SAVE IR4
         CLA      2,4
         STA      IOXEF            SET EOF EXIT
         ARS      18
         STA      IOXER            SET PERMANENT REDUNDANCY EXIT
         CAL      1,4              GET FIRST WORD OF CALL SEQUENCE
         STT      IOXSLL           SAVE MODE FOR SELECT
         STP      IOXSLL           SAVE PREFIX FOR SELECT TYPE
         STA      IOXND            PUT LOC IN ACTIVE CALLING SEQUENCE
         STA      *+1              SET TO PICK UP UCB
         LAC      **,4             -L(UCB)
         ARS      18               I/O COMMANDS LOC TO ADDRESS
         STA      IOXSLL           PUT IN SELECT WORD
         CLA      IOXSLL           GET SELECT WORD
         ZET      1,4              TEST FOR OTHER USE OF THIS UNIT
         TRA      *-1              WAIT TILL UNIT FREE
         STO      1,4              SIGNED CONTROL WORD TO UCB WORD 2
         STZ      IOXIN            SET IN-OPERATION WORD ON
         TSX      (ACTIV,4         GO TO ACTIVATE
IOXND    PZE      **               UNIT
         NZT      IOXIN            TEST FOR REQUEST COMPLETE
         TRA      *-1              NOT DONE, WAIT
         LDI      IOXIN            PICK UP COMPLETION BITS SET BY IOXSEL-
IOXS4    AXT      **,4             SET FOR EXIT
         LFT      200000           TEST EOF, EOT
IOXEF    TRA      **               EOF EXIT
         LFT      100000           TEST FOR PERM. REDUNDANCY
IOXER    TRA      **               YES, ERROR EXIT
         TRA      3,4              NORMAL RETURN
*
IOXSLL PZE        0,,IOXSEL        LOCATION OF SELECT ROUTINE
IOXIN    PZE      **               IN OPERATION CELL
IOXMD    PZE      **               MODE SWITCH
*
*        IOXSEL ROUTINE ENTERED TWICE BY IOEX FOR EACH I/O OPERATION
*
IOXSEL SXA        IOXSS4,4         SAVE IR 4
         PAC      0,4              -L(UCB)
         TMI      IOXPST           SELECT MINUS OR POSTING ENTRY
         CLA      0,4              UCB WORD 1
         PDX      0,2              UNIT TO IR 2
         CLA      1,4              UCB WORD 2
         STA*     (RCHXI           STORE LOC TO RCHX
         STT      IOXMD            SAVE MODE FLAG
         NZT      IOXMD            TEST MODE
         TXI      *+1,2,16         SET BINARY MODE FOR UNIT
         TMI      IOXWR            WRITE
         SXA      *+1,2            PLACE READ SELECT ADDRESS
         RDS      **               READ SELECT
         XEC*     (RCHXI           ISSUE RESET LOAD CHANNEL
IOXSS4 AXT        **,4             RESTORE IR 4
         TRA      1,4              AND EXIT IOXSEL
IOXWR    SXA      *+1,2            PLACE SELECT ADDRESS FOR WRITE
         WRS      **               WRITE SELECT
         TRA      IOXSS4-1         GO TO ISSUE CHANNEL COMMANDS
IOXPST STI        IOXIN            POSTING ENTRY   SAVE ERROR FLAGS
         STL      IOXIN            SET  IN-OPERATION WORD OFF
         STZ      1,4              SET UCB WORD 2 ZERO
         TRA      IOXSS4           GO TO EXIT
```

Appendix C: Bit Assignments of 7631 Sense Data Words

SNSDTA



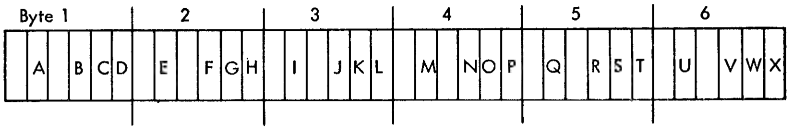
SNSDTA+1



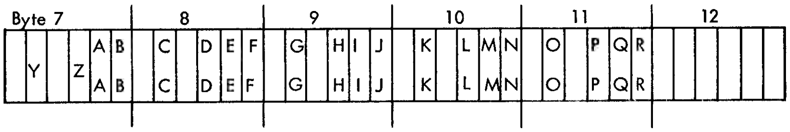
Bit	Assignment	Comment
A	Not Used	Summary Bits
B	Program Check	
C	Data Check	
D	Exceptional Condition	
E	Invalid Sequence	Program Check
F	Invalid Code	
G	Format Check	
H	No Record Found	
I	Invalid Address	Program Check for CE Track
J	Response Check	Data Check
K	Data Compare Check	
L	Parity or Cyclic Code Check	
M	Access Inoperative	Exceptional Condition
N	Access Not Ready	
O	Disk/Drum Circuit Check	
P	Control Unit Circuit Check	
Q	Channel Interrupt	
R	Six-Bit Mode	
S	Reserved	
T	Reserved	
U	ACC-MOD 00	Attention
V	ACC-MOD 01	
W	ACC-MOD 02	
X	ACC-MOD 03	
Y	ACC-MOD 04	
Z	ACC-MOD 05	
AA	ACC-MOD 06	
BB	ACC-MOD 07	
CC	ACC-MOD 08	
DD	ACC-MOD 09	

Appendix D: Bit Assignments of 7640 Sense Data Words

SNSDTA



SNSDTA+1



Bit	Assignment	Comment
A	Operator Required	Summary Bits
B	Program Check	
C	Data Check	
D	Exceptional Condition	
E	X	Selected Tape
F	X	Drive Number
G	X	(E + G = Drive 0)
H	X	
I	Drive Not Ready	Operator Required
J	Drive Not Loaded	Program Check
K	Drive File Protected	
L	Not Used	
M	Invalid Order	
N	Drive Busy	
O	Drive at BOT	
P	Drive at EOT	
Q	Correction Occurred	Data Check
R	Parity Check	
S	Code Check	
T	Envelope Check	
U	Overrun Check	
V	Excessive Skew Check	
W	Track Start Check	
X	Not Used	
Y	Read a Tape Mark	Exceptional Condition
Z	Drive in EWA	
AA	No Data Transmitted	
BB	Not Used	
CC	Read Section Busy	
DD	Write Section Busy	
EE	Backward Mode	
FF	Not Used	
GG	Drive 0	
HH	Drive 1	
II	Drive 2	
JJ	Drive 3	
KK	Drive 4	
LL	Drive 5	
MM	Drive 6	
NN	Drive 7	
OO	Drive 8	
PP	Drive 9	
QQ	Not Used	
RR	Not Used	

Appendix E: Sample Scatter-Read Program (Disk)

```
   THE FOLLOWING EXAMPLE IS SOLELY FOR THE PURPOSE OF ILLUSTRATION.
     IT DOES NOT REFLECT BUFFER TECHNIQUES IN USE OF 7909 CHANNEL.

*       SAMPLE ROUTINE   PERFORM SCATTER-READ FROM DISK USING IOEX
*
START   LAC     SYSLB1,1        -L(UCB) TO IR1.
        CLA     SYSTCH          SEEK LOCATION NEXT RECORD.
        STO     FLAG            SAVE IT IN FLAG.
        ARS     18
A       ZET     1,1             UCB 2 FREE...
        TRA     *-1             NO, WAIT.
        STA     3,1             PLACE IN UCB 4 FOR IOEX TO DO SEEK.
        CLS     2,1             SET AF FLAG SO IOEX WILL SEEK.
        STO     2,1
        TSX     (ACTIV,4        PERFORM SEEK WHILE SETTING
        PZE     SYSLB1            UP ORDER, ETC.
*
* UCB  2 CONTENTS ARE ZERO.  A SEEK REQUEST MAY BE PLACED WITHOUT
* PRIORITY PENDING.
*
AA      CLA     FLAG            PICK UP CURRENT SEEK A/O VERIFY
*                               BINARY ADDRESS.
        TSX     (DECVD,4        CONVERT ADDRESS TO BCD
        LAC     SYSLB1,4        -L(UCB) TO IR4.
        TSX     (FDAMT,2        BYTES 3-6 OF MQ, TTTT IN BCD.
        BCI     1,0CBM06        HA2 IDENTIFIER IS BM.
        PZE     DVTA            LOCATION OF ORDER TO BE COMPOSED.
        CLA     9IOSL           SET UCB 2 PRIORITY.
        STO     1,1
        TSX     (ACTIV,4        PERFORM READ.
        MZE     SYSLB1
        ZET     1,1             WAIT TILL DONE.
        TRA     *-1
        LDI     9IOSLI          7909 CONDITION PLACED BY (SEL-).
        LFT     001000          WAS IT UNUSUAL END...
        TRA     (PAWSX          YES.  READ IOEX COMMENTS ON PRINTER.
                                GO TO ERROR PAUSE.
        CLA     FLAG            NO.  MORE OF THIS IBSYS RECORD...
        TMI     SYSTRA          NO.  GO TO IBSYS CONTROL.
        PDX     0,4             PICK UP NEXT SEEK ADDRESS.
        PXA     0,4             DOES NEXT TRACK IN BINARY REQUIRE
        LRS     35              A SEEK...
        DVP     =40
        TNZ     AA              NO.  SEEK NOT NEEDED UNLESS
        PXA     0,4             REMAINDER IS ZERO.
        TRA     A+2             YES.  SET PRIORITY FOR SEEK TO IOEX.
*
* CONSTANTS
*
9IOSL   PZE     CTLR,,9SL       PRIORITY WORD UCB 2.
FLAG    PZE
DVTA    DVTA    ,,,             VERIFY ORDER COMPOSED BY (FDAMT.
9IOSLI  PZE                     IOEX GIVEN 7909/7631 STATUS
*
* ONE PURPOSE IOEX TYPE SELECT ROUTINE
*
9SL     TRA     SE
        PZE     *+1,,1          EXCLUSION WORD ADDRESS,,COUNT.
        OCT     062001000000    EXCLUSION MASK BITS.
* ERROR PROCEDURE EXCLUDED.  STATUS RESET ON MATCHED EXCLUSION
* BITS FOR ERROR INTERRUPTS.
SE      PAC     0,2             -L(UCB).
        TMI     SM              SELECT MINUS.
        CLA     1,2             UCB2.
        STA*    (RCHXI          SET RSCX ADDRESS, VIA IR1.
        CLA     9SL+1           SET EXCLUSION WORD (OPTIONAL).
        SLW*    (RCTXI          (OPTIONAL).
        XEC*    (RCHXI          RESET AND START CHANNEL.
        TRA     1,4             EXIT TO IOEX.
SM      STZ     1,2             RESET PRIORITY WORD FOR THIS ACCESS.
        STI     9IOSLI          SAVE 7909/7631 STATUS.  ALL ERRORS
*                               ARE INDICATED BY IOEX EVEN IF
*                               EXCLUSION FLAGS ARE USED.
        TRA     1,4             EXIT TO IOEX.
*
* 7909 PROGRAM
*
CTLR    SMS*    AA+4            INHIBIT ATTN AND SELECT INTERFACE
        CTLR    DVTA            DVTA AND SET READ STATUS.
        CPYP    FLAG,,1         FIRST WORD OF A TRACK.  ITS
        CPYP    *+1,,1          DECREMENT USED BY CPU PROGRAM.
        WTR     *               SIMULATE SCATTER-LOAD FROM
        TCH     *-2             TAPE ON DISK.
* PROGRAM NOT TERMINATED BY TWT BECAUSE A TRAP WILL OCCUR DUE
* TO 7909 SEQUENCE ERROR CONDITION, WHEN A CPYD IS LOADED INTO
* WORD  CTLR+4.
```

Appendix F: Use of Input/Output Units

IBSYS Operating System

The symbolic unit-reference structure of the 7090/7094 IBSYS Operating System is intended to allow the installation more flexibility in the use of the available input/output units. Certain input/output units are assigned to System Unit functions, and the remainder are made available for use by object programs. In the charts that follow, the term "Disk" refers to IBM 1301 and 2302 Disk Storage Units. The following chart shows the over-all requirements of the 7090/7094 IBSYS Operating System:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X	X	X			X
System Input Unit (SYSIN1)	Yes	X	X		X		
System Output Unit (SYSOU1)	Yes	X	X				
System Peripheral Punch (SYSPP1)	Yes	X	X				
System Card Reader (SYSCRD)	Optional				X		
System Printer (SYSPRT)	Yes					X	
System Utility Units 1-4 (SYSUT1-4)	Yes	X	X	X			X
System Utility Units 5-9 (SYSUT5-9)	No	X	X	X			X
System Checkpoint Units 1-2 (SYSCK1-2)	No	X	X	X			X

The use of cylinders of a direct access storage unit as one of the system units requires a 7631 File Control with the cylinder-mode feature.

IBJOB Processor

The following chart specifies the input/output unit requirements of the IBJOB Processor:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X	X	X			X
System Input Unit (SYSIN1)	Yes	X	X		X		
System Output Unit (SYSOU1)	Yes	X	X				
System Peripheral Punch (SYSPP1)	Yes	X	X				
System Card Reader (SYSCRD)	Optional				X		
System Printer (SYSPRT)	Yes					X	
System Utility Units 1-4 (SYSUT1-4)	Yes	X	X	X			X
System Utility Units 5-9 (SYSUT5-9)	No	X	X	X			X
System Checkpoint Unit 1 (SYSCK1)	No	X	X	X			X
System Checkpoint Unit 2 (SYSCK2)	*	X	X	X			X

*The System Checkpoint Unit 2 (SYSCK2) is required if load-time debugging is requested.

Symbolic Update Program

The following chart specifies the input/output unit requirements of the Symbolic Update Program:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X		X			X
System Input Unit (SYSIN1)	Yes	X					
System Output Unit (SYSOU1)	Yes	X					
System Peripheral Punch (SYSPP1)	No						
System Card Reader (SYSCRD)	No						
System Printer (SYSPRT)	Yes					X	
System Utility Units 1-4 (SYSUT1-4)	*	X					
System Utility Units 5-9 (SYSUT5-9)	**						
System Checkpoint Units 1-2 (SYSCK1-2)	*	X					

*Any two of these units are required as Update Input and Update Output units.

**Cannot be referenced.

Disk/Drum Utilities

The following chart specifies the input/output unit requirements of the Disk/Drum Utilities:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X	X	X			X
System Input Unit (SYSIN1)	Yes	X	X	X			
System Output Unit (SYSOU1)	No	X	X				
System Peripheral Punch (SYSPP1)	No						
System Card Reader (SYSCRD)	No						
System Printer (SYSPRT)	Yes					X	

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Utility Units 1-9 (SYSUT1-9)	No	X	X	X			X
System Checkpoint Units 1-2 (SYSCK1-2)	No	X	X	X			X

9PAC Processor

The following chart specifies the input/output unit requirements of the 9PAC Processor:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X		X			X
System Input Unit (SYSIN1)	Yes	X					
System Output Unit (SYSOU1)	Yes	X					
System Peripheral Punch (SYSPP1)	*	X					
System Card Reader (SYSCRD)	Yes				X		
System Printer (SYSPRT)	Yes					X	
System Utility Unit 1 (SYSUT1)	No						
System Utility Units 2-4 (SYSUT2-4)	**	X					
System Utility Units 5-9 (SYSUT5-9)	***						

*The System Peripheral Punch is required only when the programmer requests a binary deck.

**System Utility Units 2, 3, and 4 may or may not be required, depending on the kind of application. The programmer should inform the operator of the utility units that will be required.

***Cannot be referenced.

Commercial Translator Processor

The following chart specifies the input/output unit requirements of the Commercial Translator Processor:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X		X			X
System Input Unit (SYSIN1)	Yes	X					
System Output Unit (SYSOU1)	Yes	X					
System Peripheral Punch (SYSPP1)	Yes	X					
System Card Reader (SYSCRD)	Yes				X		
System Printer (SYSPRT)	Yes					X	
System Utility Unit 1 (SYSUT1)	Yes	X					
System Utility Unit 2 (SYSUT2)	Yes	X					
System Utility Unit 3 (SYSUT3)	*	X					
System Utility Unit 4 (SYSUT4)	No						
System Utility Units 5-9 (SYSUT5-9)	**						

*System Utility Unit 3 is used as an overflow unit, and may not be used for short programs.

**Cannot be referenced.

Generalized Sorting System

The following chart specifies the input/output unit requirements of the Generalized Sorting System:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X		X			X
System Input Unit (SYSIN1)	Yes	X	X		X		
System Output Unit (SYSOU1)	Optional	X	X				
System Peripheral Punch (SYSPP1)	No						
System Card Reader (SYSCRD)	Optional				X		
System Printer (SYSPRT)	Yes					X	
At least four tape units	*	X	X				
System Utility Units 1-4 (SYSUT1-4)	**	X	X				
System Utility Units 5-9 (SYSUT5-9)	***						
System Checkpoint Unit 1 (SYSCK1)	***						
Sys Checkpoint Unit 2 (SYSCK2)	Optional		X				

*At least two tape units on each of two channels.

**System Utility Units 1-4 may be used as any of the four required tape units as specified by Sort Control cards.

***Cannot be referenced.

Input/Output Control System

The following chart specifies the input/output unit requirements of the Input/Output Control System:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X	X	X			X
System Input Unit (SYSIN1)	Yes	X	X		X		
System Output Unit (SYSOU1)	Yes	X					
System Peripheral Punch (SYSPP1)	No	X					
System Card Reader (SYSCRD)	No				X		
System Printer (SYSPRT)	Yes					X	
System Utility Units 1-3 (SYSUT1-3)	No	X	X	X			X

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Utility Unit 4 (SYSUT4)	Yes	X	X	X			X
System Utility Units 5-9 (SYSUT5-9)	*						
System Checkpoint Units 1-2 (SYSCK1-2)	No	X	X	X			X

*Cannot be referenced.
NOTE: The Restart Program's input/output requirements are the same as those for the Input/Output Control System.

FORTRAN II Processor

The following chart specifies the input/output unit requirements of the FORTRAN II Processor:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	Disk	711	716	7320
System Library Unit (SYSLB1)	Yes	X		X			X
System Input Unit (SYSIN1)	Yes	X					
System Output Unit (SYSOU1)	Yes	X					
System Peripheral Punch (SYSPP1)	Yes	X					
System Card Reader (SYSCRD)	No				X		
System Printer (SYSPRT)	Yes					X	
System Utility Units 1-4 (SYSUT1-4)	Yes	X					
System Utility Units 5-9 (SYSUT5-9)	*						
System Checkpoint Unit 1 (SYSCK1)	**	X					
System Checkpoint Unit 2 (SYSCK2)	No	X					

*Cannot be referenced.
**Required if the SNAPSHOT feature of IOP is utilized.

System Editor

The following chart specifies the input/output unit requirements for the System Editor:

SYSTEM UNIT FUNCTION	REQUIRED	CAN BE: 729	7340	1301	711	716	7320
System Library Unit (SYSLB1)	Yes	X	X	X			X
System Input Unit (SYSIN1)	Yes	X	X		X		
System Output Unit (SYSOU1)	Optional	X	X				
System Peripheral Punch (SYSPP1)	No	X	X				
System Card Reader (SYSCRD)	No				X		
System Printer (SYSPRT)	Yes					X	
System Utility Unit 1 (SYSUT1)	Yes	X	X	X			X
System Utility Unit 2 (SYSUT2)	*	X	X	X			X
System Utility Unit 3 (SYSUT3)	**	X	X	X			X
System Utility Units 4-9 (SYSUT4-9)	No	X	X	X			X
System Checkpoint Units 1-2 (SYSCK1-2)	No	X	X	X			X

*Required if alteration cards are not on SYSIN1.
**Required for creating split systems using the SYSALT option.
NOTE: A second System Library Unit (SYSLB2) is required if editing from an alternate unit is specified.

Appendix G. System Monitor and System Editor Messages

This appendix lists in alphabetical order the error messages that are printed by the System Monitor and System Editor. An asterisk (*) to the right of a message indicates that the message is printed on- and off-line. Additional information on the System Monitor messages can be found in the publication *IBM 7090/7094 IBSYS Operating System; Version 13: Operator's Guide*, Form C28-6355.

System Monitor Messages

CONTROL CARDS NEEDED IN CARD READER
OPER. ACTION PAUSE

Explanation: The System Monitor has detected the end-of-file condition in the card reader while attempting to read a control card.

1EOR MOUNT NEW SYSOU1 *
OPER. ACTION PAUSE

Explanation: The program has written over the end-of-reel reflective spot on the System Output Unit.

ILLEGAL SYSUN1 DEFINITION *
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTION PAUSE

Explanation: One of the following conditions has occurred:

1. The system unit specified on either a $RELEASE or $AS card cannot be found in the System Unit Name Table.
2. The starting cylinder or number of cylinders specified on a $AS card contains a nonnumeric character.
3. One or both of the system units specified on a $SWITCH card cannot be found in the System Unit Name Table.
4. The system unit specified on a $REWIND, $ENDFILE, $REMOVE, $PROTECT, or $UNLOAD card cannot be found in the System Unit Name Table.

ILLEGAL UNIT SPECIFIED *
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTION PAUSE

Explanation: The control card printed on-line before the message is a $ATTACH or $DETACH card with an invalid unit specification in the variable field.

ILL UNIT REQ'ST AT xxxxx

Explanation: A calling sequence to IOEX from octal location xxxxx has requested an operation on an invalid unit.

$\left\{\begin{matrix}\text{INTF CK}\\\text{SEQ CK}\end{matrix}\right\}$
PRES START TO GO ON

Explanation: An interface check (INTF CK) or a sequence check (SEQ CK) on a 7909 data channel has been detected.

I/O CK
PRES STRT TO GO ON

Explanation: An input/output check on 7607 or 7909 data channel has been detected.

xxxxxx NO ASSIGNMENT MADE *
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTION PAUSE

Explanation: One of the following conditions has occurred:

1. The library unit on which the System Monitor expected to find a particular subsystem is unassigned.
2. The unit from which the System Monitor should read control cards (according to the setting of sense switch 1) is unassigned.

NO SYSIN1 EITHER

Explanation: The System Input Unit has not been assigned. This message is issued in conjunction with the message SYSCRD NO ASSIGNMENT MADE S.S.1 SETTING IGNORED

NOT A BASIC MONITOR CONTROL CARD *
PUSH START TO INGORE OR
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTION PAUSE

Explanation: The System Monitor has read a control card that it cannot recognize. The control card itself is printed just before the message.

OUTPUT END OF REEL

Explanation: The end-of-reel reflective spot has been encountered on the output tape.

$RESTART IGNORED—ABSOLUTE VALUE OF COUNT *
TOO LARGE
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTION PAUSE

Explanation: One of the following conditions has occurred:

1. If a $RESTART +n card has been used, this message indicates that a $STOP card was encountered before the system had spaced forward n jobs.
2. If a $RESTART −n card has been used, this message indicates that the beginning-of-tape marker was encountered before the system had backspaced n jobs.

$RESTART IGNORED—ILLEGAL FIELD *
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTON PAUSE

Explanation: The System Monitor has read a $RESTART card with an invalid character in the variable field.

$RESTART IGNORED—NO MATCH FOUND *
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTION PAUSE

Explanation: There is no $JOB card on the System Input Unit with an identification that corresponds to that of the $JOB card that follows the $RESTART MATCH card just read.

SPILL TAPE READ ERROR

Explanation: A read error has occurred on the overflow tape, but the dump will proceed without a halt to ensure at least a partial recovery of the contents of core storage and to allow the system to continue. If this procedure is not satisfactory for a particular application, the job should be rerun.

SYSCRD NO ASSIGNMENT MADE
S.S.1 SETTING IGNORED

Explanation: The card reader is unassigned. An automatic switch to the System Input Unit is made.

SYSLBx NOT TAPE (LB1 IS)—CANT $EXECUTE *

Explanation: The $EXECUTE card has requested a subsystem on an alternate library unit (SYSLBx). If this is a disk or drum unit and if IBSYS resides on tape, this request is invalid.

SYSOU1 IS NOT ASSIGNED. NO DUMP CAN BE TAKEN *

Explanation: The System Output Unit has not been assigned; therefore, a dump cannot be taken at this time.

SYSPP2 IS NOT ASSIGNED, NO DUMP CAN BE TAKEN *

Explanation: The System Peripheral Punch has not been assigned; therefore, a dump cannot be taken.

SYSPP2 IS NOT 729 OR HYPER. NO DUMP CAN BE TAKEN *

Explanation: The alternate System Peripheral Punch is neither a 729 Magnetic Tape Unit nor a 7340 Hypertape Drive; therefore, a dump cannot be taken at this time.

xxxxxx UNKNOWN SYSTEM *
PROVIDE CLARIFICATION IN CARD READER
OPER. ACTION PAUSE

Explanation: The $EXECUTE card that is printed before this message specified a program that the System Monitor could not find on the System Library Unit.

UNIT xx yyyy ERROR
$m_1 m_2 m_3 \ldots m_{24}$

Explanation: A read or write error has occurred on the Hypertape drive with the symbolic address xx. The symbolic address is the same as that used on the $ATTACH card. The letters yyyy are either READ or WRITE. The m_n are 24 octal characters of sense data received from the file control. There is no pause associated with this message.

UNIT xx yyy ERROR-TRK 00zzzz
$m_1 m_2 m_3 \ldots m_{24}$

Explanation: A read or write error has occurred on the disk or drum unit with symbolic address xx. The symbolic address is the same as that used on the $ATTACH card. The letters yyyy are either READ or WRITE. The letters zzzz stand for the track address of the last seek performed by IOEX. The m_n are 24 octal characters of sense data received from the file control. There is no pause associated with this message.

UNIT xx $\left\{ \begin{array}{l} \text{NTRDY} \\ \text{NTLOD} \\ \text{FILPR} \end{array} \right\}$
OPER. ACTION PAUSE

Explanation: The condition specified prevents the program from reading or writing with the specified Hypertape drive. The symbols are as follows:

1. NTRDY – Hypertape drive xx is not ready.
2. NTLOD – Hypertape drive xx is not loaded.
3. FILPR – Hypertape drive xx is file protected.

UNIT xx REC. yyyyy FILE zzzzz 25 ERASES ON WRITE

Explanation: The program has tried to write record yyyyy in file zzzzz on unit xx, and has failed 26 times.

UNIT xx REC. yyyyy FILE zzzzz – NOISE ON ERASE

Explanation: An error occurred while the program was trying to erase an area of magnetic tape.

UNIT xx REC. yyyyy FILE zzzzz PERM. READ REDUN.

Explanation: An uncorrectable error occurred in reading record yyyyy of file zzzzz from unit xx.

UNIT xx REC. yyyyy FILE zzzzz REC. DISCRDED-NOISE

Explanation: Record number yyyyy in file number zzzzz on unit xx has been discarded because it was a noise record.

UNIT xx REC. yyyyy FILE zzzzz WROTE SHORT REC

Explanation: Record yyyyy of file zzzzz on unit xx is less than three words in length and will therefore be indistinguishable from a noise record when it is read.

UNIT xx SEEK UN END
PRES STRT TO GO ON

Explanation: An unusual ending has resulted from a search (SEEK) of the disk or drum unit with symbolic address xx.

System Editor Messages

A SYSUN1 IS MISSING. OPERATION
CARDS NOT PUNCHED

Explanation: The requested HYPERTAPE or DISK LOAD card cannot be punched because neither the card punch nor the System Peripheral Punch is assigned. Edit is completed.

CARD COUNT ERROR-GIVEN COUNT
xxxxxx ACTUAL xxxxxx

Explanation: The number appearing on the *CHECK card does not correspond to the actual number of modification cards.

CHANGE CARD xxxxxx-BINARY CARD
OUT OF ORDER†

Explanation: An editor control card or end-of-file was expected at this point. The xxxxxx is the card sequence number.

CHANGE CARD xxxxxx-CHECKSUM ERROR *
EDITING CONTINUES

Explanation: The checksum appearing on a column binary alteration card does not match the one compared by the Editor. The xxxxxx is the card sequence number.

CHANGE CARD xxxxxx-INCORRECT
EDITOR CONTROL CARD†

Explanation: Self-explanatory. The xxxxxx is the card sequence number.

EDITING BYPASSED–ERROR IN *
PREVIOUS JOB SEGMENT

Explanation: Self-explanatory.

EDITING BYPASSED xxxxxx UNUSED *
MODIFICATION CARDS

Explanation: This message follows one or more diagnostics indicating the type of error that caused the edit to be discontinued.

EDITING CONTINUES AFTER PERM.
ERROR 729/1301

Explanation: A redundancy was encountered when reading from System Library Unit 1 or System Library Unit 2.

EDITING NOT COMPLETED BECAUSE *
OF ERROR

Explanation: This message is written in conjunction with the messages that are marked with a †.

EDITING IS DISCONTINUED DUE TO
UNRECOVERABLE ERROR AT xxxxx

Explanation: An error has occurred while reading or writing disk. The xxxxx is the track number associated with the error.

EDITOR HAS DETECTED AN ERROR IN
ORDER OF FILES

Explanation: This message is provided at the completion of an edit in which the diagnostic "THE xxxxxx SYSTEMS POSITION DOES NOT AGREE WITH THE SYSNAM TABLE" has been issued.
NOTE: This message is printed off-line.

END OF TAPE WHILE WRITING SYSTEM
TAPE†

Explanation: Self-explanatory.

†This message is written in conjunction with the message EDITING NOT COMPLETED BECAUSE OF ERROR

EOT ON SYSOU1. NEW OUTPUT TAPE NEEDED

Explanation: Self-explanatory.

ERROR . . . IF SYSTEM ON DISK, IBSYS MUST BE ON DISK TOO†

Explanation: One of the following conditions has occurred.

1. System Library Unit 1 must be on disk if System Library Unit 2 is on disk (in a System Library Unit 2 edit).
2. System Utility Unit 1 must be on disk if System Utility Unit 3 is on disk (in an edit in which SYSALT is specified).

ERROR IN I/O UNIT†

Explanation: System Input Unit or System Output Unit is assigned as disk or drum.

ERROR . . . RECORD NAME NOT FOUND IN SYSTEM NAME TABLE†

Explanation: In a System Library Unit 2 edit, the name of the first record on System Library Unit 2 must be one of the following:

1. IBSYS, if SYSLB2 is disk, or 5U0002 (the name of the IBSYS Load Tape record), if SYSLB2 is tape.
2. Any system record name that appears in the System Name Table.

FILEOR *PLACE REQUEST TOO LATE. CARD IGNORED . . .

Explanation: The tape has already been spaced past the record name or system name specified on a FILE *AFTER card or a *PLACE card.

HYPERTAPE INPUT REDUNDANCY

Explanation: An I/O check, sequence check, or interface check has occurred while reading the System Input Unit when the input unit is a 7340 hypertape.

ILLEGAL EOF RETURN†

Explanation: A relocatable binary card image was expected at this point.

IMPROPER INSERT CARD, SKIP TO NEXT CONT. CARD

Explanation: The second parameter on an *INSERT card is not SYSALT. Editing continues at the next control card.

IMPROPER $ BINARY CONTROL * CARD SEQUENCE†

Explanation: A control card in a relocatable binary modification deck is out of order. The correct order is: $IBLDR, $TEXT, $CDICT, $DKEND.

INVALID BINARY CARD IN RELOCATABLE PROCESSING

Explanation: Self-explanatory.

INVESTIGATE CHANNEL TROUBLE. START

Explanation: An I/O check, sequence check, or interface check has occurred while reading or writing disk or hypertape.

MODIFICATION FILE REDUNDANCY

Explanation: A redundancy has been encountered during reading of the modification file on System Utility Unit 2.

SOME SYSUN1 HAS NO UNIT†

Explanation: The system unit function specified on an Editor control card is not assigned.

SYSALT OPERATION IS NOT ALLOWED WITH LB2 EDIT†

Explanation: Self-explanatory.

SYSLDR DICTIONARY TOO LARGE†

Explanation: The number of records to be written onto disk exceeds the number that can be recorded in the System Loader Table.

SYSUN1 CYL. LIMIT REACHED. SS4 DOWN, GO ON. UP, END EDIT

Explanation: The last cylinder assigned to a system unit function has been reached while writing on a disk or drum.

TAPE IDENTIFICATION ERROR. xxxxxx WAS CHECK WORD. xxxxxx IS TAPE ID

Explanation: The tape ID that is specified on the *CHECK card does not match that which appears in the *EOT record.

TAPE REDUNDANCY

Explanation: A redundancy has been encountered during reading of the modifications from the System Input Unit.

THE xxxxxx SYSTEMS POSITION DOES NOT AGREE WITH THE SYSNAM TABLE

Explanation: The record name specified was encountered at a point where a system name was expected. NOTE: This message is printed off-line.

UNRECOGNIZED PARAMETER ON *EDIT CARD-xxxxxx

Explanation: Self-explanatory. xxxxxx is the parameter.

$DKEND CARD MISSING† *

Explanation: A control card is missing from a relocatable binary modification deck.

*CHECK CARD VARIABLE FIELD ERROR. CARD IGNORED

Explanation: Self-explanatory.

*PLACE ERROR. IBSYS POSITION TABLE FULL. INSERT IGNORED

Explanation: There is no room for inserting another name in the 50-entry System Name Table.

*PLACE ERROR. IMPROPER FIELD OR REQUEST–CARD IGNORED

Explanation: Self-explanatory.

†This message is written in conjunction with the message EDITING NOT COMPLETED BECAUSE OF ERROR

Index

IBM Technical Newsletter

File Number	7090-36
Re: Form No.	C28-6248-3,-4,-5,-6,-7
This Newsletter No.	N28-0242
Date	December 20, 1967
Previous Newsletter Nos.	N28-0207 N28-0182 N28-0187 N28-0137 N28-0166 N28-0139

IBM 7090/7094 IBSYS Operating System, Version 13: System Monitor (IBSYS)

This Technical Newsletter amends the IBM Systems Reference Library publication IBM 7090/7094 IBSYS Operating System, Version 13: System Monitor (IBSYS), Form C28-6248-3,-4,-5,-6,-7.

In the referenced publication, replace the pages listed below with the corresponding pages attached to this newsletter:

Pages	Subject of Amendment
75,76	Figure 58. 7320 Update-Edit Deck Number 2

The symbol (•) to the left of the figure caption indicates that the figure has been revised.

File this cover letter at the back of the publication as confirmation that all changes have been received and incorporated into the publication.

IBM Corporation, Programming Systems Publications, 1271 Avenue of the Americas, New York, N.Y. 10020

PRINTED IN U.S.A.

When this is necessary, a special modification deck is distributed. However, the cards from these special decks should not be incorporated in the Cumulative Editor Deck. In particular, see the first modification letter issued to the IBJOB Subroutine Library after the release of a new version for details concerning the IBLIB Cumulative Editor Deck.

The user may keep a copy of the distributed System Library Tape as a backup tape and use it and the Cumulative Editor Deck whenever it is necessary to produce a System Library Tape at the current modification level. If a special modification deck is distributed, it may be used to produce a new backup System Library Tape from the old one. The new backup tape and the Cumulative Editor Deck may then be used, when necessary, to produce an up-to-date System Library Tape.

Maintaining a Two-Tape System Library

Some installations may use a two-tape System Library, in which parts of the IBJOB Processor are located on a second System Library Tape. To incorporate IBM modifications without changing the distributed modification decks, a duplicate of the System Library should be maintained on a single tape reel. After modifications are incorporated in the single System Library Tape, it may be used to produce a two-tape System Library, in which parts of the IBJOB Processor are located on a second tape. A discussion of the use of multiple library units is contained in the publication *IBM 7090/7094 IBSYS Operating System: IBJOB Processor,* Form C28-6389.

7320 Drum Update-Edit Decks for IBJOB

The 7320 Drum Update-Edit Decks are designed to serve as a guide for providing IBJOB system residence on 7320 Drum Storage. These decks are shown in Figures 57 and 58.

Edit Deck Number 1 splits the IBSYS Operating System, on a 729 Disk/Drum/Hypertape-Capability Sys-

```
$JOB            7320 EDIT DECK NO. 1
$ATTACH         CNAM/S
$AS             SYSUT1,010,000
$ATTACH         A4
$AS             SYSUT3,H
$IBEDT
       *EDIT    MAP,MODS,CNAM/S
       *PLACE   UPDATE
       *PLACE   SORT
       *PLACE   IBSFAP
       *PLACE   FORTRA
       *PLACE   DK9OUT
       *PLACE   RESTAR
       *PLACE   CT
       *PLACE   9PAC
       *PLACE   IOCS
       *PLACE   CIFSR,2,2,1
       *PLACE   UPDATE,1,2,2
       *PLACE   SORT,2,2,3
       *PLACE   IBSFAP,1,2,4
       *PLACE   FORTRA,5,2,5
       *PLACE   DK9OUT,1,2,6
       *PLACE   RESTAR,1,2,7
       *PLACE   CT,10,2,8
       *PLACE   9PAC,4,2,9
       *PLACE   IOCS,3,2,10
       *MODIFY  IBSYS
15141  *OCT     400001000001            A1 IS SYSLB2
       *MODIFY  IBJOB                   CHANGE ACTION LIST FOR IBLIB
11117  *OCT     200000000000
11122  *OCT     200001000000
21335  *OCT     200000000000
       *AFTER   IBLDRO
       *DUP     SYSLB1,SYSALT,1         COPY OFF CIFSR
       *INSERT  FILEMK
       *AFTER   IBLDRQ
       *DUP     SYSLB1,SYSALT,1         COPY OFF TIFSR
       *INSERT  FILEMK
       *AFTER   IBLDRS
       *AFTER   FILEMK
       *DUP     SYSLB1,SYSALT,28        COPY OFF REST OF SYSTEMS
'       END-OF-FILE CARD
$IBSYS
$PAUSE          CHANGE A4 TO A1, CLEAR CORE,LOAD DISK LOAD CARD.
```

Figure 57. 7320 Update-Edit Deck Number 1

```
$JOB            7320 EDIT DECK NO. 2
$ATTACH         A3
$AS             SYSCK1
$ATTACH         B3
$AS             SYSCK2
$REWIND         SYSCK1
$REWIND         SYSCK2
$REWIND         SYSPP1
$EXECUTE        UPDATE
        UPDATE  9,10
DRUM    EQU     1                                                          2M012200
STJOB2  BCI     9,$ETC    CONT D   IOEX,MAXBUF,BLK=524,CORE=500000000000   2M165420
DRUM    EQU     1                                                          2O012200
DRUM    EQU     1                                                          2P012200
        CAL*    UIIC41                  PATCH FOR VERSION 5             (1)2P462800
        END     -1                                                         2S530500
        ENDFIL  10
        REWIND  10
        UNLOAD  9
        ENDUP
$IBSYS
$PAUSE          PLEASE MOUNT SCRATCH TAPE ON A3.
$EXECUTE        IBJOB
$IBJOB IBLDR    NOGO
$IEDIT          SYSCK2,SRCH
$IBMAP IBLDRM   9000,ABSMOD,JOBSYM,()OK,M94,LNREF
$IBMAP IBLDRO   4000,ABSMOD,JOBSYM,()OK,M94,LNREF
$IBMAP IBLDRP   11000,ABSMOD,JOBSYM,()OK,M94,LNREF
        END-OF-FILE CARD
$IBSYS
$ENDFILE        SYSPP1
$REWIND         SYSPP1
$REMOVE         SYSCK2
$PAUSE          PLEASE MOUNT SCRATCH TAPE ON B3.
$SWITCH         SYSPP1,SYSUT2
$SWITCH         SYSUT3,SYSUT1
$*      A3 WILL BECOME FINAL SYSTEM TAPE.
$IBEDT
      *EDIT     MAP,MODS
TAPE  *REPLACE  IBLDRM
TAPE  *REPLACE  IBLDRN
TAPE  *REPLACE  IBLDRO
TAPE  *REPLACE  IBLDRP
TAPE  *REPLACE  IBLDRQ
        END-OF-FILE CARD
$IBSYS
$REMOVE         SYSLB1
$ATTACH         A4
$AS             SYSLB1
$ATTACH         A3
$AS             SYSUT1
$ATTACH         B3
$AS             SYSUT3
$EXECUTE        IBJOB
$IBJOB          EDIT,LOGIC,NOSOURCE
        END-OF-FILE CARD
$IBSYS
$IBEDT
      *EDIT     MAP,MODS
FILE  *REPLACE  CIFSR,SYSUT4
FILE  *REPLACE  TIFSR,SYSUT4
        END-OF-FILE CARD
$STOP
```

• Figure 58. 7320 Update-Edit Deck Number 2

tem Library Tape, between a 7320 drum and a 729 System Tape. The IBSYS System Monitor, the IBJOB Monitor (minus the IBJOB Subroutine Library (IBLIB)), and the IBSYS Editor are edited onto a 7320 drum, while the IBJOB Subroutine Library and all other subsystems under the IBSYS Operating System are edited onto a 729 tape. This 729 System Library Tape (SYSLB2) is used in conjunction with those systems edited onto the 7320 drum. Edit Deck Number 1 accomplishes this split of the IBSYS Operating System with no assembly or reblocking of components required.

Octal modification cards are included in deck number 1 for the IBSYS and IBJOB records to modify the system to do the following:

1. Attach tape unit A1 as SYSLB2.
2. Modify the IBJOB ACTION table to indicate that IBLIB is on SYSLB2.

A one-pass IBEDT is performed. During this edit, SYSLB2 is created on tape unit A4. A map of the system as it resides on 7320 drum and on a 729 tape is written on the system output tape. At the end of the

C28-6248-7

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Printed in U.S.A. C28-6248-7